Approches hybrides pour la recherche sémantique de l'information : intégration des bases de connaissances et des ressources semi-structurées

Yassine Mrabet

Approches hybrides pour la recherche sémantique de l'information : Intégration des bases de connaissances et des ressources semi-structurées

Ecole doctorale Informatique de Paris-Sud

THÈSE

de

L'UNIVERSITÉ PARIS-SUD

présentée pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ PARIS-SUD

Spécialité : Informatique

par

Yassine Mrabet

Rapporteurs :	Patrice Buche	INRA SUPAgro, Montpellier	(Ingénieur de recherche, HDR)
	Mathieu Roche	LIRMM, Université Montpellier 2	(Maître de conférences, HDR)
Examinateurs :	Bernd Amann	LIP 6, UPMC	(Professeur)
	Yolaine Bourda	SUPELEC, (E3S)	(Professeur)
	Anne Vilnat	LIMSI, Université Paris Sud	(Professeur)
Directrice de thèse :	Chantal Reynaud	LRI, Université Paris Sud	(Professeur)
Co-directrices :	Nacéra Bennacer	SUPELEC, (E3S)	(Maître de conférences)
	Nathalie Pernelle	LRI, Université Paris Sud	(Maître de conférences)

12 juillet 2012

A mes parents qui m'ont appris à persévérer

A ma femme pour sa présence et son soutien

A ma fille pour avoir su nous donner le sourire

A tous ceux qui m'ont soutenu durant ce parcours

Remerciements

Je voudrais tout d'abord remercier les membres du jury pour avoir accepter d'examiner ce travail de thèse et pour leurs précieux retours.

Je tiens également à remercier mes directrices de thèse : Nacéra Bennacer, Nathalie Pernelle et Chantal Reynaud pour leur disponibilité et leurs efforts tout au long de ce travail.

Je remercie aussi tous les membres de l'équipe IASI et l'équipe du département Informatique à Supélec pour leur accueil et pour les échanges très intéressants que l'on a eus durant ces années.

Enfin, je voudrais remercier tous ceux sans qui cette thèse n'aurait pas été possible et en particulier ma femme Asma, mon oncle Hamouda, mon frère Mehdi et mes parents pour leur précieux soutien tout au long de ce parcours.

Résumé

La recherche sémantique de l'information a connu un nouvel essor avec les nouvelles technologies du Web sémantique. Des langages standards permettent aujourd'hui aux logiciels de communiquer par le biais de données écrites dans le vocabulaire d'ontologies de domaine décrivant une sémantique explicite. Cet accès "sémantique" à l'information requiert la disponibilité de bases de connaissances décrivant les instances des ontologies de domaine. Cependant, ces bases de connaissances, bien que de plus en plus riches, contiennent relativement peu d'information par comparaison au volume des informations contenues dans les documents du Web.

La recherche sémantique de l'information atteint ainsi certaines limites par comparaison à la recherche classique de l'information qui exploite plus largement ces documents. Ces limites se traduisent explicitement par l'absence d'instances de concepts et de relations dans les bases de connaissances construites à partir des documents du Web. Dans cette thèse nous étudions deux directions de recherche différentes afin de permettre de répondre à des requêtes sémantiques dans de tels cas. Notre première étude porte sur la reformulation des requêtes sémantiques des utilisateurs afin d'atteindre des parties de document pertinentes à la place des faits recherchés et manquants dans les bases de connaissances. La deuxième problématique que nous étudions est celle de l'enrichissement des bases de connaissances par des instances de relations.

Nous proposons deux solutions pour ces problématiques en exploitant des documents semi-structurés annotés par des concepts ou des instances de concepts. Un des points clés de ces solutions est qu'elles permettent de découvrir des instances de relations sémantiques sans s'appuyer sur des régularités lexico-syntaxiques ou structurelles dans les documents. Nous situons ces deux approches dans la littérature et nous les évaluons avec plusieurs corpus réels extraits du Web. Les résultats obtenus sur des corpus de citations bibliographiques, des corpus d'appels à communication et des corpus géographiques montrent que ces solutions permettent effectivement de retrouver de nouvelles instances relations à partir de documents hétérogènes tout en contrôlant efficacement leur précision.

Abstract

Semantic information retrieval has known a rapid development with the new Semantic Web technologies. With these technologies, software can exchange and use data that are written according to domain ontologies describing explicit semantics. This "semantic" information access requires the availability of knowledge bases describing both domain ontologies and their instances. The most often, these knowledge bases are constructed automatically by annotating document corpora. However, while these knowledge bases are getting bigger, they still contain much less information when comparing them with the HTML documents available on the surface Web.

Thus, semantic information retrieval reaches some limits with respect to "classic" information retrieval which exploits these documents at a bigger scale. In practice, these limits consist in the lack of concept and relation instances in the knowledge bases constructed from the same Web documents. In this thesis, we study two research directions in order to answer semantic queries in such cases. The first direction consists in reformulating semantic user queries in order to reach relevant document parts instead of the required (and missing) facts. The second direction that we study is the automatic enrichment of knowledge bases with relation instances.

We propose two novel solutions for each of these research directions by exploiting semi-structured documents annotated with concept instances. A key point of these solutions is that they don't require lexico-syntactic or structure regularities in the documents. We position these approaches with respect to the state of the art and experiment them on several real corpora extracted from the Web. The results obtained from bibliographic citations, call-for-papers and geographic corpora show that these solutions allow to retrieve new answers/relation instances from heterogeneous documents and rank them efficiently according to their precision.

Table des matières

Table des figures

Chapitre 1

Introduction

Contexte et problématique

La recherche d'information à partir de documents numériques a connu une évolution fulgurante avec le World Wide Web. Le challenge posé aux premiers moteurs de recherche du marché dans les années 90 n'est plus comparable aux défis posés aujourd'hui. Le nombre de documents publics disponibles sur Internet (ou Web de surface[1]) dépasse les 11.5 milliards de pages depuis 2005 [Gulli & Signorini, 2005] et continue de croître de manière exponentielle.

Face à ce grand volume d'information, les moteurs de recherche industriels se sont investis pour leur plus grande partie dans les méthodes de traitement statistique de l'information. Dans ce contexte, le mot est considéré comme une simple chaîne de caractères dont la fréquence d'apparition des occurrences est déterminante dans la sélection des documents retournés aux utilisateurs.

La recherche sémantique de l'information offre une vision différente dans laquelle le mot n'est plus seulement une simple chaîne de caractères mais aussi une référence à des concepts ou à des d'entités du monde réel ou à des relations entre ces entités. Dans le cadre du Web sémantique[Berners-Lee *et al.*, 2001; Shadbolt *et al.*, 2006] des langages standards du W3C[2] tels que RDF(S), OWL et SPARQL ont été définis pour représenter des données conformément à un vocabulaire défini dans des ontologies, raisonner sur ces données et les interroger.

Dans ce contexte, l'interrogation sémantique se fait au travers de requêtes SPARQL formulées suivant le vocabulaire d'une ontologie, couvrant généralement un domaine d'application particulier. Les bases de connaissances RDF(S)/OWL inter-

1. Partie du Web indexable par les moteurs de recherche
2. World Wide Web Consortium, http://www.w3.org

rogées représentent des informations sous la forme (sujet,relation,objet) et permettent d'obtenir des réponses de différents types :

– des concepts et des relations définis dans l'ontologie
– des instances de concepts
– des instances de relations
– des valeurs littérales correspondant à des valeurs d'attributs d'instances

Les faits des bases de connaissances interrogées sont généralement construits automatiquement par l'annotation de documents. Ainsi de nombreux outils d'extraction ou d'annotation ont aujourd'hui pour objectif d'annoter les documents du Web par des concepts et des relations définis dans des ontologies ou par des instances définies dans des bases de connaissances.

De plus en plus de bases de connaissances RDF(S)/OWL volumineuses sont aujourd'hui publiées sur le Web (e.g. DBpedia, Yago, GeoNames), notamment dans le cadre du projet Linked Open Data[1]. Cependant, ces bases de connaissances, bien que très volumineuses, contiennent relativement peu d'information par comparaison au volume d'informations contenues dans les documents textuels. Une des raisons principales de ce fait est que les approches d'annotation, qui sont à la base de la construction et de l'enrichissement des bases de connaissances, s'appuient sur l'existence de régularités lexico-syntaxiques ou structurelles dans les documents. Or ces régularités ne sont pas toujours présentes dans les documents du Web, caractérisés par leur très grande hétérogénéité. Aussi, en excluant les documents du Web qui ne comportent pas de régularités exploitables, l'interrogation sémantique se prive de ressources importantes en volume et riches en terme de contenu.

Objectifs et contributions

À travers cette thèse, nous proposons des solutions pour améliorer la recherche sémantique de l'information en exploitant les documents semi-structurés du Web ne présentant pas de régularités lexico-syntaxiques ou structurelles exploitables. Plus précisément, notre principal objectif consiste à :

1. Obtenir de nouvelles réponses à des requêtes basées sur le vocabulaire d'une ontologie de domaine et recherchant des relations entre des instances de concepts.
2. Trier ces réponses suivant leur qualité.

1. http://www.linkeddata.org/

Cette problématique est étudiée dans deux cas de figure différents :

1. Un premier cas où nous disposons uniquement de documents annotés par des concepts du domaine. Nous proposons pour cela une première approche, appelée **SHIRI** [1]**-Querying**, qui reformule les requêtes sémantiques posées pour rechercher des parties de documents référant aux instances de concepts recherchées et aux relations exprimées dans la requête de l'utilisateur.
2. Un deuxième cas où nous disposons (i) de documents annotés par des instances de concepts du domaine et (ii) de bases de connaissances préexistantes contenant des faits caractérisant des instances du domaine, dont celles référencées par les documents annotés. Nous proposons pour cela une approche, appelée **REISA** (cont**R**oled **E**xtension and **I**nterrogation of **S**emantic **A**nnotations), qui enrichit les bases de connaissances avec de nouvelles relations entre instances de concept et permet d'interroger l'ensemble des connaissances (enrichies ou préexistantes) de manière transparente à l'utilisateur.

Première contribution : SHIRI-Querying

Notre première contribution est une approche de reformulation des requêtes utilisateurs visant à interroger sémantiquement des documents semi-structurés annotés par des concepts du domaine. Nous nous plaçons ici dans les deux cas fréquents où (i) les annotateurs ne se sont pas intéressés à retrouver précisément les instances de concept et ont annoté les documents avec les concepts uniquement ou (ii) les annotateurs ont créé des instances de concept pendant l'annotation mais ne leur ont associées aucune description à part leur concept. Les requêtes sémantiques des utilisateurs n'auront ainsi aucune réponse dans leur forme initiale car soit les instances de concepts ne sont pas identifiées soit elles ne sont décrites par aucune instance de relation.

Dans les travaux actuels, les critères utilisés pour la reformulation des requêtes utilisateurs se basent essentiellement sur la structure et la sémantique des ontologies et n'exploitent pas les documents annotés. Mis à part le fait que ces approches ne permettent pas de retrouver des réponses si les instances de concepts et de relations ne sont pas créées ou identifiées, ce type de reformulation ne tient pas compte des contextes de niveau document obtenus par l'annotation.

Nous montrons à travers *SHIRI-Querying*[Mrabet *et al.*, 2009, 2010a,b] que le contexte d'occurrence des (instances de) concepts dans un document peut (i) jouer un rôle important dans le processus de reformulation de requêtes et (ii)

1. Projet Emergence DIGITEO SHIRI (http://wwwdi.supelec.fr/~bennacer/SHIRI.htm/)

apporter de nouvelles réponses en retournant des parties de document qui réfèrent à des relations entre des instances de concepts, même si ces dernières ne sont pas explicitement identifiées dans une base de connaissances.

Pour découvrir des relations entre instances de domaine dans un tel cas nous exploitons les relations de voisinage entre les nœuds des documents XHTML (nœuds de l'arbre DOM). Ces nœuds sont annotés en utilisant les annotations de leur contenu textuel et les règles SHIRI-Annot[Thiam *et al.*, 2009] qui permettent de caractériser les nœuds suivant leur hétérogénéité sémantique.

Dans une seconde étape, pour retrouver des nœuds de documents référant aux instances de concepts et de relations recherchés par la requête utilisateur, nous avons défini des reformulations fondées sur (i) les relations de voisinage entre les nœuds dans le document et (ii) l'hétérogénéité sémantique des nœuds en termes de concepts. Nous avons proposé un algorithme, appelé **DREQ** (**D**ynamic **R**eformulation and **E**xecution of **Q**ueries), qui construit et exécute dynamiquement l'ensemble des reformulations. Cette construction respecte une relation d'ordre que nous avons définie entre les requêtes reformulées dans le but de trier les réponses finales suivant l'hétérogénéité sémantique des nœuds. Cet algorithme a été implémenté et expérimenté sur deux corpus réels collectés à partir du Web. Les résultats que nous avons obtenus montrent bien que l'approche (i) permet d'atteindre des nœuds de document contenant des occurrences des instances de domaine et des relations requises par la requête initiale de l'utilisateur et (ii) que la relation d'ordre définie permet effectivement de donner la priorité aux réponses les plus précises.

Deuxième contribution : REISA

Notre deuxième contribution porte sur l'enrichissement des bases de connaissances avec des instances de relations. L'approche que nous proposons dans ce cadre, *REISA*, vise à découvrir des instances de relations à partir des documents semi-structurés pour compléter les données des bases de connaissances préexistantes interrogées[Mrabet *et al.*, 2012]. Une des caractéristiques importantes de cette approche est qu'elle est applicable même en l'absence de régularités lexico-syntaxiques ou structurelles dans les documents exploités. Cet enrichissement exploite la distance de voisinage entre les occurrences des instances de concepts dans le document pour découvrir de nouvelles instances de relation candidates qui sont contrôlées grâce :

– aux axiomes de fonctionnalité (inverse) des relations et aux instances de relations déjà présentes dans les bases de connaissances.
– aux règles de domaine spécifiques à certaines relations (e.g. la capitale d'un pays doit faire partie de celui-ci géographiquement)

– au Web pour filtrer les candidats en soumettant une expression textuelle de l'instance de relation candidate via un moteur de recherche.

Les nouvelles instances de relations sont regroupées dans une base de connaissances spécifique appelée base d'enrichissement. Cette base d'enrichissement est une base de connaissances RDF(S)/OWL interrogée au même titre que les bases de connaissances préexistantes mais ayant une mesure de confiance différenciée. Cette mesure de confiance tient compte de la distance entre les parties de document qui ont permis la production des nouvelles instances de relations. D'autres mesures de confiance sont aussi associées aux bases de connaissances préexistantes par des experts du domaine. *REISA* permet à l'utilisateur d'interroger l'ensemble de ces bases de connaissances d'une manière transparente tout en triant les réponses suivant le poids des faits retournés.

REISA a été implémentée et expérimentée sur deux corpus extraits du Web et des bases de connaissances issues du projet Linked Open Data [1]. Une première expérimentation a été effectuée sur un corpus d'appels à communications pour évènements scientifiques avec les bases de connaissances DBLP RDF [2] et WKB [3]. Une deuxième expérimentation a été effectuée sur un corpus du domaine géographique extrait de Wikipedia avec la base de connaissances DBpedia [4]. Les résultats montrent que :

1. ne pas s'appuyer sur des régularités de structuration ou d'expressions permet d'augmenter considérablement le rappel (i.e. nombre d'instances de relations découvertes).
2. l'exploitation des bases de connaissances préexistantes permet de contrôler efficacement les instances de relations candidates et d'atteindre une bonne précision. Cette précision est plus prononcée si l'ontologie de domaine exploitée définit peu de propriétés différentes entre deux concepts donnés.
3. cette performance est améliorée par la validation Web des instances de relations candidates.

1. http://www.linkeddata.org
2. http://thedatahub.org/dataset/l3s-dblp
3. http://www.ontotext.com/kim/ontologies
4. http://www.dbpedia.org

Plan du manuscrit

Chapitre 2. État de l'art : annotation sémantique de documents semi-structurés et reformulation de requêtes sémantiques

Ce chapitre présente un état de l'art sur les deux grandes directions de recherche que nous avons mentionnées plus haut, à savoir, la production et l'enrichissement des connaissances et la relaxation ou reformulation de requêtes. Nous y présentons d'abord les principaux types des travaux d'annotation sémantique automatique des documents semi-structurés. Ces travaux comprennent aussi bien les approches d'annotation par concept ou instance de concepts que les approches de découverte d'instances de relations sémantiques. La dernière partie de ce chapitre sera dédiée au résumé des principaux travaux de reformulation des requêtes sémantiques et au positionnement de nos approches.

Chapitre 3. Modèle Sémantique d'Intégration

Ce chapitre est dédié (i) à la présentation modèle sémantique d'intégration **SIM** (**Semantic Integration Model**) qui est exploité par les deux approches proposées dans cette thèse et (ii) à la formalisation des problématiques étudiées. Nous présentons d'abord le modèle SIM qui permet de représenter de manière homogène les entités de documents, les bases de connaissances et les annotations des documents. Nous introduisons ensuite les définitions préliminaires RDF et SPARQL nécessaires pour la formalisation du type de requêtes que nous traitons. La dernière partie de ce chapitre est consacrée à la formalisation des deux cas de figure étudiés.

Chapitre 4. Reformulation de requêtes pour l'interrogation sémantique de documents semi-structurés

Dans ce chapitre nous présentons notre approche pour l'interrogation sémantique de documents semi-structurés annotés, *SHIRI-Querying*, qui se fonde sur la reformulation des requêtes utilisateurs pour retourner des nœuds de document pertinents. La première partie du chapitre est dédiée à la présentation de l'annotation des nœuds de document qui est effectuée suivant les règles d'annotation de *SHIRI-Annot*. Dans une deuxième partie de ce chapitre, nous décrivons notre approche de reformulation qui exploite des opérations de reformulation élémentaires et les annotations des nœuds de document pour construire plusieurs requêtes reformulées. Cette construction suit une relation d'ordre définie pour trier les reformulations de la requête utilisateur et retourner les réponses les plus précises d'abord. Dans la dernière partie de ce chapitre, nous présentons l'algorithme *DREQ* qui exploite les reformulations élémentaires et la relation d'ordre que nous avons définis pour construire et exécuter dynamiquement toutes les reformulations de la requête

utilisateur.

Chapitre 5. Évaluation et synthèse de l'approche *SHIRI-Querying*

Ce chapitre est dédié à l'évaluation de l'approche *SHIRI-Querying* sur deux corpus réels extraits du Web. La précision des réponses retournées par *SHIRI-Querying* est évaluée en fonction de la distance de voisinage entre les nœuds dans le document et en fonction de la relation d'ordre définie entre les reformulations. Dans la dernière partie de ce chapitre nous présentons une discussion et une synthèse de l'approche *SHIRI-Querying* à la lumière des résultats obtenus.

Chapitre 6. Enrichissement contrôlé de bases de connaissances à partir de documents semi-structurés

Dans ce chapitre, nous décrivons l'approche *REISA*. Nous présentons successivement l'architecture générale de l'approche, le processus d'enrichissement comprenant les différents contrôles par le voisinage, les bases de connaissances et le Web. Nous décrivons enfin la méthode d'interrogation que nous proposons pour exploiter en même temps les connaissances préexistantes, issues de l'annotation ou produites par notre approche d'enrichissement, tout en tenant compte de la confiance accordée à chacune de ces sources.

Chapitre 7. Évaluation et synthèse de l'approche *REISA*

Dans ce dernier chapitre, nous évaluons l'approche d'enrichissement *REISA* sur deux corpus réels extraits du Web. Nous présentons d'abord les résultats obtenus sur un premier corpus d'appel à communications pour des conférences scientifiques. Nous présentons ensuite les résultats obtenus avec un deuxième corpus portant sur le domaine géographique, extrait de Wikipedia. Enfin, nous proposons une discussion et une synthèse de l'approche *REISA* à travers l'analyse de ces résultats.

Nous concluons et donnons quelques perspectives à nos travaux dans le dernier chapitre.

Chapitre 2

État de l'art : Annotation sémantique de documents semi-structurés et reformulation de requêtes sémantiques

La recherche sémantique d'information repose sur l'exploitation d'annotations sémantiques des sources d'information accessibles. Une annotation sémantique est une information additionnelle associée à un document qui précise le sens du document ou d'une de ses parties. L'existence d'annotations sémantiques modifie

la recherche d'information. Il ne s'agit plus seulement de rechercher des chaînes de caractères mais d'exploiter des données décrivant le contenu sémantique des documents et permettant de répondre à une requête utilisateur structurée. Dans le cadre du Web sémantique, ces informations additionnelles utilisent un vocabulaire particulier défini formellement dans une ontologie. La recherche sémantique consiste alors à répondre à des requêtes formulées suivant ce même vocabulaire.

Afin d'obtenir plus de réponses à ces requêtes, deux directions différentes peuvent être explorées. La première direction consiste à enrichir les connaissances disponibles en utilisant les documents semi-structurés. La seconde direction consiste à proposer des modifications de requêtes (e.g. des reformulations, approximations ou relaxations) afin d'atteindre des réponses inaccessibles à partir de la requête utilisateur initiale.

Nous commençons cet état de l'art en examinant la première direction de recherche qui consiste à enrichir des bases de connaissances préexistantes en utilisant les documents semi-structurés. Nous nous intéressons principalement à la tâche de découverte d'instances de propriétés et à la tâche de découverte d'instances de concepts comme un prérequis de la première tâche. Nous présentons tout d'abord les caractéristiques de l'annotation sémantique. Nous décrivons ensuite successivement les problèmes posés par la découverte d'instances de concepts, puis ceux posés par la découverte d'instances de relations sémantiques, en donnant quelques exemples d'approches et de méthodologies proposées pour les résoudre. La deuxième partie de cet état de l'art porte sur la reformulation des requêtes utilisateur pour atteindre plus de réponses. Dans la dernière partie nous positionnons l'ensemble de nos travaux.

2.1 Annotation sémantique de documents

Une annotation est une information additionnelle associée à un document ou à une partie de document pour ajouter une connaissance, un lien vers une autre ressource ou une question. L'**annotation sémantique** est une annotation qui ajoute une connaissance formalisée de façon explicite dans une ontologie.

Une **ontologie** est une spécification d'une conceptualisation [Gruber, 1993, 1995]. Dans une publication plus récente [Gruber, 2008] propose une définition plus détaillée des ontologies que nous traduisons comme suit :

> « Une ontologie est un ensemble de primitives de représentation qui permettent de modéliser un domaine de connaissances ou un domaine de discours. Les primitives de représentation sont typiquement des classes (ou des ensembles), des attributs (ou des propriétés), et des relations (ou relations parmi les membres des classes). La définition des primitives de représentation inclut des informations à propos de leur sens et des contraintes sur la consistance logique de leur application »

Dans cet état de l'art nous appellerons **base de connaissances** une ontologie peuplée par des individus : instances de concepts, de relations et d'attributs.

Les approches d'annotation sémantique se distinguent par les **types de connaissances** ontologiques utilisés pour l'annotation et par la nature et la granularité des parties de document annotées. Les éléments ontologiques utilisés pour l'annotation peuvent être classés en cinq catégories :

- Les concepts (e.g. le terme "protein" peut référer à un concept défini dans une ontologie biomédicale).
- Les instances de concepts (e.g. l'expression "ONU" peut réfèrer à une instance du concept *Organisation*).
- Les propriétés (e.g. l'expression "est situé à" réfère à la propriété *localisationDe*)
- Les instances de propriétés (e.g. l'expression "Paris est située en France" réfère à une instance de la propriété *localisationDe*)
- Les valeurs littérales (e.g. "1945" est une valeur possible de date).

Les **entités de documents** annotées sémantiquement peuvent être :
- des documents entiers (e.g. [Bizer *et al.*, 2009; Suchanek *et al.*, 2008])
- des parties structurées d'un document (e.g. [Hignette *et al.*, 2009; Thiam *et al.*, 2009]) tels les nœuds de l'arbre DOM d'un document XHTML ou des sections encadrées par des balises spécifiques dans un document HTML.
- des groupes nominaux (e.g. [Khelif & Dieng-Kuntz, 2004])
- des groupes nominaux spécifiques tels que les entités nommées (e.g. [Cimiano *et al.*, 2005; Popov *et al.*, 2004]) : l'expression *entité nommée* (EN) est apparue lors de la conférence MUC-6 (Message Understanding Conference) [Grishman & Sundheim, 1996]. L'emploi de l'adjectif nommé a pour objectif de définir les entités qui ont un désignant déterminé (e.g. "France Telecom", "John Dœ"). Cela inclut les noms propres ou les expressions telles que les noms d'espèces (e.g. "cèdre du Liban"), de maladies, ou de substances chimiques. Cette définition a été élargie aux expressions temporelles telles que les dates et les heures, ou à des valeurs numériques (e.g. "0.12 Kg/L"). Les catégories d'entités nommées qui sont le plus souvent considérées sont les personnes, les lieux, les organisations (catégories ENAMEX de la conférence MUC) et les expressions temporelles. Des hiérarchies d'EN peuvent aussi être définies.

```
<div about="http ://dbpedia.org/resource/Albert_Einstein" rel="dbp:citizenship">
   <span about="http ://dbpedia.org/resource/Germany"></span>
   <span about="http ://dbpedia.org/resource/United_States"></span>
</div>
```

FIGURE 2.1 – Exemple d'annotations en *RDFa*

La **sérialisation** des annotations sémantiques peut se faire dans le document lui-même ou en dehors du document dans un ficher spécifique. Par exemple, des annotations au format RDF pourront être exprimées sous forme de micro-formats [1] ou d'attributs RDFa [2] dans le document lui même, ou dans des fichiers séparés sérialisés en *RDF-XML* ou *N-TRIPLE/N3*. Par exemple, le triplet RDF

<kb:TimBernersLee> <*foaf:page*> "en.wikipedia.org/wiki/Tim_Berners_Lee"

lie l'instance du concept *Person* <kb:TimBernersLee> et la page Wikipedia "http://en.wikipedia.org/wiki/Tim_Berners_Lee" par un lien de référence.

Dans un autre exemple, le segment de document HTML de la figure 2.1 lie l'instance du concept *Person* correspondant à "Albert Einstein" à des nationalités par la propriété *dbp:citizenship*.

2.2 Découverte de concepts et d'instances de concepts

La découverte de concepts et d'instances de concepts dans les documents donne lieu à une annotation si le lien entre l'entité de document annotée et le concept (ou l'instance de concept) est conservé. Ces deux tâches sont intrinsèquement liées car la découverte d'une instance de concept implique indirectement la découverte du concept de cette instance. De plus, dans l'autre sens, la majeure partie des approches qui annotent les documents par un concept de domaine expriment le fait qu'une (ou des) référence(s) à des instances de ce concept ont été localisées dans l'entité de document annotée [Abdelhamid *et al.*, 2009; Sereno *et al.*, 2005; Thiam *et al.*, 2009]). Une exception est à noter pour les approches qui visent à construire des ontologies où les documents servent effectivement à découvrir de nouveaux concepts. Ces approches ne sont cependant pas le sujet de notre

1. http ://microformats.org/get-started
2. http ://www.w3.org/TR/rdfa-syntax/

étude qui porte uniquement sur l'annotation de documents avec des ontologies préalablement définies.

La découverte d'instances de concepts consiste à retrouver des entités de document référant à des instances de concepts définis dans une ontologie. Quand une base de connaissances est disponible, il s'agit de retrouver des références à des instances répertoriées dans la base de connaissances et éventuellement de créer les instances à référencer si elles ne sont pas répertoriées. Cette découverte peut donner lieu à une annotation si le lien entre l'entité de document et l'instance est conservé.

Les approches de découverte d'instances de concepts peuvent être classifiées selon les méthodes utilisées (e.g. apprentissage, "pattern matching") et les ressources exploitées mais aussi suivant la granularité et le type des entités de document annotées.

• Les approches basés sur des **techniques d'apprentissage** sont capables d'inférer des règles de découverte d'instances de concepts à partir d'un corpus d'entraînement constitué de textes annotés. De tels systèmes utilisent, par exemple, les caractéristiques lexico-syntaxiques des exemples positifs et éventuellement négatifs ainsi que les caractéristiques de leur contexte (e.g. [Mendes *et al.*, 2011; Suchanek *et al.*, 2006]).

• Certaines approches exploitent des **patrons lexico-syntaxiques** génériques ou définis par un expert du domaine d'application (e.g. CPankow [Cimiano *et al.*, 2005], KIM [Popov *et al.*, 2004], SOFIE [Suchanek *et al.*, 2009]). Par exemple, KIM utilise des patrons lexico-syntaxiques pour associer une instance, désignée par une entité nommée, à un concept. Ainsi, c'est à partir du mot clé "valley" que "Barossa valley" sera reconnu comme réfèrant à une instance du concept *Valley*. La figure 2.2 présente quelques exemples de patrons permettant l'extraction d'entités nommées. Il est à noter que de nombreuses approches permettent d'extraire et de catégoriser les entités nommées sans pour autant référer à une ontologie (e.g. [Bikel *et al.*, 1997], [Nadeau & Sekine, 2007]).

• Plusieurs approches exploitent des **ressources lexicales**, comme des listes d'entités nommées (e.g. listes Wikipedia, noms de personnes de DBLP [1][Hassell *et al.*, 2006]) ou des composants lexicaux associés à des ontologies de domaine (e.g. la composante lexicale de l'ontologie de *SHIRI-Extract*[Thiam *et al.*, 2009]) ou à des bases de connaissances (e.g. base d'alias de la base de connaissances de *KIM*[Popov *et al.*, 2004], labels des instances dans la base de connaissances pour le système *SOFIE* [Suchanek *et al.*, 2009]). Par exemple, *KIM* reconnait les expressions référant aux instances de concept définies dans sa base de connaissances

1. http ://www.informatik.uni-trier.de/~ley/db/

Patron (expressions régulières en PERL)	Description
\w∗ (\w city) \w∗.	Extrait des noms de villes constitués d'un seul mot précédé et suivi par zéro ou plusieurs autres mots.
\w∗ the film director is ([A−Z][A−Za−z]{1,10}){1,3} \w∗.	Extrait des noms de directeurs de films constitués de 1 à 3 mots de 2 à 11 caractères dont le premier est obligatoirement une majuscule. Il est suivi et précédé par zéro ou plusieurs autres mots.

FIGURE 2.2 – Exemple de patrons d'extraction d'entités nommées

grâce aux attributs *alias* décrivant différentes expressions pouvant référer à une instance dans un document (e.g. "N.Y.", "N.Y.C" ou "New York" sont trois alias de la ville de New York).

Ce type d'approche permet de pallier certains cas de variations syntaxiques des expressions pouvant référer à une même instance. Cependant, un autre problème réside dans la délimitation des expressions dans les textes à annoter. Ainsi, dans ([Khelif & Dieng-Kuntz, 2004]) des fenêtres textuelles de 4 mots au plus sont utilisées pour reconnaître des instances de concepts dans le domaine biomédical. Les différentes séquences de 4 mots de chaque phrase sont soumises au processus de catégorisation : le système vérifie si les séquences extraites correspondent à des termes d'un thésaurus du domaine [1] sous leur forme initiale ou tronquée (i.e. sous parties constituées de 3 mots puis 2 puis 1). Lorsqu'une correspondance est trouvée, la séquence de mots associée à l'instance est identifiée comme étant la sous partie la plus longue qui a pu être catégorisée. Cependant, le seuil de mots fixé pour la fenêtre reste discutable car certaines entités dépassent les 4 mots, comme indiqué dans un des exemples utilisés pour leur travaux avec la fonction cellulaire "alveolar epithelial type II cell proliferation" qui compte 6 mots.

• Les **bases de connaissances** peuvent aussi constituer une ressource sémantique pour la découverte d'instances de concepts. Par exemple, le système SOFIE ([Suchanek *et al.*, 2009]) utilise la technique des *sacs de mots* pour comparer les expressions extraites avec les labels des instances d'une base de connaissances. Plus précisément, la comparaison se fait entre deux sacs de mots. Pour un terme t_i dans un document D et une instance i, le premier sac de mots est constitué de t_i et des mots-clés du document D. Le deuxième sac de mots contient les labels de l'instance i et les labels de ses instances voisines dans la base de connaissances (i.e. instances reliées directement à i par une relation sémantique). La valeur de similarité est calculée comme étant la proportion normalisée de l'intersection entre les mots des deux sacs. Les valeurs de similarités sont ensuite utilisées dans

1. Le méta-thésaurus utilisé est l'UMLS, http://www.nlm.nih.gov/research/umls/

un système de résolution afin de déterminer l'instance de concept la plus proche du terme t_i.

• Dans une approche similaire [Hassell *et al.*, 2006] désambiguïse les noms de personnes grâce à leurs co-auteurs, décrits dans la base de connaissances *DBLP-RDF*[1].

• Un autre exemple d'approches exploitant les bases de connaissances est DBpedia Spotlight [Mendes *et al.*, 2011] qui permet à un utilisateur d'annoter des expressions apparaissant dans des documents textuels en utilisant les IRIs[2] des instances de concepts définies dans la base de connaissances DBpedia. Une première étape permet de sélectionner les expressions du texte susceptibles de référer à une ressource de DBpedia en recherchant les labels d'instances de classes DBpedia. Plusieurs instances peuvent être candidates (e.g. l'expression "Washington" peut référer à dbpedia :Georges_Washington, dbpedia :Washington,_D.C. ou dbpedia :Washington_(U.S._state)). Afin de sélectionner l'instance appropriée, DBpedia Spotlight utilise les paragraphes des articles Wikipedia dans lesquels apparaissent des liens vers d'autres pages Wikipédia. Une page Wikipedia ayant souvent une instance de concept correspondante dans DBpedia, ces paragraphes constituent des contextes d'utilisation qui sont représentés sous forme de vecteurs de mots. Un pouvoir discriminant est calculé pour chaque couple (mot, instance) afin d'indiquer la pertinence d'un mot pour caractériser le contexte de référence textuelle à une instance donnée. Une fois les pouvoirs discriminants calculés, la découverte de l'instance de concept se fait en comparant le contexte de l'entité de document à annoter au vecteur de mots associé à l'instance. Cette comparaison permet aussi à DBpedia SpotLight d'associer une mesure de confiance aux annotations proposées.

• Les ressources lexicales ou les bases de connaissances sont cependant souvent incomplètes, voire absentes pour certains domaines, ce qui conduit d'autres approches à utiliser des ressources ouvertes et évolutives comme le Web. Ainsi, CPankow extrait des entités nommées candidates grâce à des patrons lexico-syntaxiques génériques puis utilise le Web pour reconnaître leur catégorie [Cimiano *et al.*, 2005]. Le procédé employé construit des requêtes mots-clés en utilisant les entités nommées et des patrons de Hearst [Hearst, 1992] (e.g. "X such as New York", "X such as Microsoft"), requêtes qui sont soumises à des moteurs de recherche tels que Google. Cela permet de récupérer des hyperonymes pouvant correspondre à des labels de concepts à partir des extraits de documents pertinents retournés par les moteurs de recherche. La pertinence des extraits de document retournés est évaluée en les comparant avec le contexte textuel des

1. http://thedatahub.org/dataset/fu-berlin-dblp
2. Internationalized Resource Identifier, http://tools.ietf.org/html/rfc3987

entités nommées à annoter. Les hyperonymes sélectionnés sont considérés comme des catégories possibles de l'entité nommée.

• Le système SHIRI-Extract utilise à la fois une ontologie à composante lexicale et des appels au Web. Si le groupe nominal/EN extrait n'est pas suffisamment similaire à un groupe nominal/EN de l'ontologie, la catégorisation se fait via le Web avec le même principe que celui utilisé dans CPankow. Les termes obtenus de ces appels Web sont comparés aux termes de l'ontologie grâce à l'outil d'alignement TaxoMap ([Hamdi *et al.*, 2009]). Si ces termes sont proches d'un terme décrivant un concept de l'ontologie, ce concept est considéré comme étant la catégorie de l'EN ou du groupe nominal. Les termes du Web, les nouvelles EN et les groupes nominaux extraits viennent ensuite enrichir la composante lexicale de l'ontologie.

Les approches d'annotation par instances de concept se fondent, pour leur plus grande partie, sur des procédés automatiques fiables. Diverses expérimentations et challenges internationaux font état de mesures de précision très élevées (supérieures à 90%) pour l'extraction et la classification des entités nommées [Nadeau & Sekine, 2007]. Le tableau 2.1 décrit les principales caractéristiques des travaux que nous avons présentés ainsi que les résultats de leur évaluation sur différents corpus.

Système	Méthode		Ressources			EDoc	Précision	Rappel
	app.	pat.	BC	BL	Web			
KIM		✓	✓	✓		EN	86%	82%
CPankow		✓			✓	EN	76%	-
SHIRI-Extract		✓			✓	EN Termes	91,6%% 74,3%	79,6% 70,8%
Spotlight	✓		✓			EN	80,5%	-
Sofie	✓	✓	✓	✓		EN	94,6%	-

Abréviations :
Pat. : méthodes à base de patrons.
App. : Apprentissage.
BC : Bases de connaissances.
BL : Bases Lexicales.
EDoc : Entités de Document.

TABLE 2.1 – Caractéristiques des outils d'annotation par instances de concepts

Ces résultats sont relativement bons et laissent penser qu'il est possible de se fonder sur les systèmes d'annotation par instances de concepts pour la découverte d'instances de relations sémantiques.

Dans la prochaine section, nous étudions la tâche de découverte de relations sé-

mantiques à partir de documents semi-structurés dans la littérature. Nous nous intéressons plus particulièrement à l'extraction d'instances de relations définies dans des ontologies.

2.3 La découverte d'instances de relations sémantiques

Dans cette partie de l'état de l'art, nous nous intéressons principalement à la découverte d'instances de relations sémantiques à partir de documents XHTML. Le terme découverte est utilisé au lieu du terme annotation car le lien entre les entités de documents et les instances de relations découvertes ne sont pas toujours conservés. Il est aussi à noter que les approches de découverte de nouvelles relations entre concepts dans le but de construire/enrichir des ontologies (e.g. [Aussenac-Gilles & Jacques, 2006; Kamel & Aussenac-Gilles, 2009]) ne sont pas concernées par cette étude.

À titre d'exemple, la phrase "*Barack Obama est président des États Unis d'amérique depuis 2007*" lie les entités du monde réel "*Barack Obama*" et "*États Unis d'Amérique*" par la relation sémantique "*presidentDe*", préalablement définie dans une ontologie de domaine.

La découverte d'une relation sémantique à partir d'un document suppose de détecter les entités de document référant à des instances de concepts et de sélectionner le contexte à exploiter pour la découverte d'instances de relations (e.g. texte apparaissanr entre deux entités nommées, phrase, ensemble de phrases, partie structurée d'un document).

Diverses techniques, éventuellement combinées, sont employées pour exploiter ce contexte. Certaines approches se basent sur l'existence de régularités textuelles ou structurelles qui peuvent être déclarées par un expert ou détectées par des techniques d'apprentissage automatique. Une autre direction de recherche est d'exploiter des ressources lexicales ou des connaissances du domaine pour améliorer le repérage des instances de relation dans les documents.

Nous présentons dans les sections suivantes un ensemble d'approches exploitant des techniques et des ressources différentes pour la découverte d'instances de relations sémantiques.

2.3.1 Approches à base de patrons lexico-syntaxiques

Les méthodes à base de patrons lexicaux-syntaxiques utilisent des modèles de phrases préalablement établis qu'ils comparent par la suite aux phrases contenant les entités candidates (e.g. [Agichtein & Gravano, 2000; Etzioni *et al.*, 2004; Suchanek *et al.*, 2006, 2009]). Cette opération est communément appelée "*Pattern Matching*". Les patrons, ou modèles utilisés, peuvent aussi bien se baser sur le texte de la phrase que sur sa syntaxe, souvent obtenue grâce à des analyseurs automatiques tels que Treetagger [1] ou OpenNLP [2]. La construction de tels patrons peut être manuelle ou automatique.

• Les systèmes *KnowItAll* et *Snowball* présentés respectivement dans [Etzioni *et al.*, 2004] et [Agichtein & Gravano, 2000] utilisent des listes de paires d'entités reliées par une relation donnée et considèrent les textes où co-occurrent ces entités source et cible comme un contexte possible de la relation. Ces contextes sont ensuite utilisés pour la construction des patrons lexico-syntaxiques en déduisant les expressions régulières les plus appropriées.

• *LEILA* [Suchanek *et al.*, 2006] étend cette technique en prenant aussi en compte des contre-exemples pour chaque relation. Après avoir construit les patrons lexicaux, LEILA applique un processus d'apprentissage automatique permettant de pallier les cas d'erreur (faux patrons). Un classifieur est entraîné sur l'ensemble des (contre-)exemples pour déceler les points communs les plus remarquables aux contextes détectés. Cette méthode ne comprend aucun mécanisme de désambiguïsation du sens, dans le cas où plusieurs relations sont possibles entre les entités source et cible.

• Dans une contribution plus récente, [Gerber & Ngomo, 2011] [3] proposent l'approche BOA pour extraire automatiquement des patrons de relations sémantiques en utilisant les "formes de surface" Wikipedia. Ils appellent forme de surface le texte correspondant à un hyperlien vers d'autres pages Wikipedia. Plus précisément, puisque une partie importante des pages Wikipedia est associée à des instances de concepts dans la base de connaissances DBpedia, ils considèrent que le texte correspondant à un hyperlien est une "forme de surface" de l'instance de concept associée au document pointé par cet hyperlien.

L'approche consiste, pour une relation R donnée, à prendre toutes les instances (i_1, R, i_2) puis à récupérer toutes les phrases qui contiennent au moins une forme de surface de i_1 et une forme de surface de i_2. Le texte compris entre les deux

1. http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/
2. http://incubator.apache.org/opennlp/
3. Une version plus récente de ces travaux est disponbile ici
http://svn.aksw.org/papers/2012/ESWC$_$BOA-ML-PR/public.pdf

formes de surface est ensuite exploité pour construire un patron linguistique de la relation R. Pour filtrer ces patrons, des scores de confiance sont calculés avec réseau de neurones qui prend en entrées plusieurs mesures caractérisant, par exemple, la spécificité d'un patron et sa fréquence d'occurrence. Cette approche a permis de retrouver de nouvelles relations avec une précision atteignant 92% sur un corpus anglais et 74% sur un corpus allemand extraits de Wikipedia.

Les méthodes à base de patrons permettent généralement de découvrir des instances de relation avec un haut degré de précision. Ceci est principalement dû au fait que le processus est appliqué sur des phrases et que très souvent tous les indices de la présence d'une relation se trouvent rassemblés dans une phrase. Cependant, une relation sémantique peut être exprimée par un grand nombre d'expressions textuelles [Aussenac-Gilles & Jacques, 2006]. Aussi, obtenir une bonne couverture de toutes les expressions possibles requiert un temps de travail important si les patrons sont écrits par des experts et/ou un nombre d'exemples imposant dans le cas où des techniques d'apprentissage automatique sont utilisées. Ces méthodes sont donc difficilement applicables lorsqu'on traite des documents hétérogènes construits indépendamment les uns des autres sans conventions d'écriture communes. Par ailleurs, lorsque la relation lie des entités qui ne sont pas dans la même phrase, il est difficile de les découvrir avec des patrons linguistiques. Les patrons ne sont pas non plus adaptés aux cas où les entités apparaissent dans des structures particulières et où la relation est implicite (e.g. auteurs d'un article dans une citation).

2.3.2 Approches exploitant la structure du document

D'autres approches se sont orientées vers l'exploitation de la structure des documents balisés du Web pour extraire des relations sémantiques. Dans les documents Web, les balises elles-mêmes ne sont souvent pas porteuses de sens (balises HTML) car elles ont été conçues à l'origine pour la bonne visualisation des documents. Cependant, puisqu'il est nécessaire qu'une certaine homogénéité de représentation soit adoptée pour que les documents soit bien lisibles par des humains, il reste possible, dans la pratique, de déceler et d'exploiter des régularités dans la structuration de ces documents pour en déduire des informations à un niveau plus sémantique.

À titre d'exemple, plusieurs travaux se sont intéressés aux infobox Wikipedia en tant qu'éléments de document structurés. Les infobox sont des tableaux récapitulatifs des données les plus importantes d'un article Wikipedia qui sont généralement affichées en haut à droite des articles. Chaque ligne de ces tableaux est constituée du nom d'un attribut ou d'une relation suivi d'une valeur textuelle ou

d'un lien hypertexte vers un autre document Wikipedia.

• *DBpedia* [Bizer *et al.*, 2009] et Yago [Suchanek *et al.*, 2008] sont deux bases de connaissances qui ont été construites à partir des infobox Wikipedia. DBpedia décrit près de 3.4 millions d'instances de concepts (e.g. personnes, lieux, organisations, films) par 1 trillion de triplets RDF. Yago décrit 2 millions d'instances de concepts à l'aide de 20 millions de triplets RDF. Les deux approches de construction de ces bases partent du principe que (i) chaque article Wikipedia réfère à une instance de concept particulière (ii) la deuxième colonne de l'infobox réfère à une instance de concept dans le cas où il s'agit d'un lien hypertexte ou à un littéral quand il s'agit d'une valeur textuelle et (iii) la première colonne de l'infobox indique le nom de la relation (ou de l'attribut) qui lie l'instance de concept associée au document à l'instance de concept (ou à la valeur de l'attribut) associée à la deuxième colonne de l'infobox. L'alignement entre ce nom de relation et les relations de l'ontologie est immédiat car les relations des ontologies de DBpedia et YAGO ont été construites à partir de ces infobox. Ainsi, selon ces approches, la découverte de relations est basée sur l'exploitation de la structure des infobox mais également sur le type des valeurs contenues dans les cellules de ces tableaux.

• D'autres documents du Web contiennent des structures régulières telles que les tables HTML qui ont fait l'objet de plusieurs études d'extraction de connaissances. Dans une approche récente, [Limaye *et al.*, 2010] extraient des instances de relations binaires à partir de tables HTML. Le processus employé utilise un modèle probabiliste à base de graphes pour identifier simultanément les instances de concept source et cible de la relation et la relation. Les bases de connaissances Yago et DBpedia sont utilisées comme référentiels pour les différents concepts, relations et instances de concepts. Des poids sont calculés pour représenter différentes notions, parmi lesquelles :

– l'adéquation des termes d'une cellule de tableau aux labels d'instances de concept dans les bases de connaissances (pour l'identification de(s) (l')instance(s) de concept associée(s)),
– l'adéquation des types de valeurs d'une colonne aux concepts des bases de connaissances,
– l'adéquation des concepts associés à une cellule aux domaines et co-domaines d'une relation,
– l'adéquation d'une instance de relation extraite d'une ligne des tableaux aux instances de relation présentes dans les bases de connaissances.

Cette approche a pu démontrer, entre autres, qu'un modèle probabiliste peut générer de bons résultats d'annotations sémantiques même sur un grand nombre de documents. Sur un corpus constitué de plusieurs milliers de tables regroupant

130.000 cellules, l'approche a permis d'obtenir une F-mesure moyenne de 83% pour la reconnaissance d'instances de concepts et une F-mesure de 58% pour l'extraction de relations sémantiques.

• [Hignette *et al.*, 2009] proposent également une approche non supervisée permettant d'annoter sémantiquement les tables apparaissant dans des documents HTML. Pour découvrir des instances de relations n-aires dans les lignes de la table, cette approche exploite les connaissances décrites dans une ontologie OWL : les signatures des relations, le nom des types symboliques et numériques, les contraintes décrivant les valeurs possibles des attributs symboliques, les intervalles de valeurs possibles pour les attributs numériques ou encore les unités de mesures possibles.

Les types symboliques de l'ontologie sont décrits par des taxonomies de termes. Par exemple, le type Food Product est associé à une taxonomie organisant 500 termes décrivant des aliments. Le type des colonnes symboliques de la table est découvert en comparant les termes présents dans les cellules de la colonne aux termes de l'ontologie et en comparant les termes présents dans le titre de la colonne aux noms des types symboliques décrits dans l'ontologie. Si le score final de similarité d'une colonne avec un type ne se distingue pas assez des scores obtenus pour d'autres types, le type est considéré comme inconnu. De la même façon, le type d'une colonne numérique est calculé en combinant les scores obtenus pour les contenus des cellules, en prenant en compte les unités de mesure et les intervalles de valeur, et le score obtenu en comparant le titre de la colonne au nom du type numérique.

Enfin, pour découvrir la relation sémantique n-aire représentée dans la table, les auteurs combine la similarité entre le titre de la table et le nom de la relation dans l'ontologie et la similarité entre les types trouvés pour les colonnes et la signature de la relation dans l'ontologie. Pour représenter l'incertitude des annotations finales, les auteurs utilisent les ensembles flous. Les annotations et les bases de données d'un entrepôt local peuvent ensuite être interrogées en utilisant le moteur d'interrogation MIEL [Buche *et al.*, 2008].

Sur 81 colonnes symboliques appartenant à des tables du domaine de la microbiologie, une précision de 89% et un rappel de 81% ont été obtenus. Sur 65 colonnes numériques apparaissant dans des tables traitant du risque chimiques dans les aliments ou de l'aéronautique, la précision obtenue est de 100% et le rappel est de 96%. Enfin, sur 123 relations annotées manuellement, la précision des relations reconnuesautomatiquement est de 80% et le rappel de 97%.

• *SOBA* est aussi un exemple d'approche de découverte de relations sémantiques qui exploite la structure des documents pour un domaine de spécialité [Buitelaar

& Siegel, 2006]. L'approche proposée exploite conjointement la structuration des documents et les contenus textuels des pages Web du domaine du football afin de construire et compléter une base de connaissances du domaine.

L'outil linguistique SProuT [1] est utilisé dans un premier temps pour annoter les documents. Cet outil permet de reconnaître des occurrences de personnes, de lieux et des valeurs numériques. Des patrons lexico-syntaxiques ont été ajoutées à l'outil pour permettre également la reconnaissance d'entités du domaine (e.g. entraîneur, joueur, arbitre). Les annotations linguistiques ainsi obtenues sont insérées dans le document avec des balises spécifiques. Cette nouvelle structure est ensuite interprétée en tant que triplets RDF décrivant des instances de concepts et de relations du domaine en se basant sur la structure imbriquée des éléments ajoutés par l'analyse linguistique.

• Il est important de noter que la découverte d'instances de relations à partir de la structure des documents a commencé et s'est développée dans des directions de recherche indépendantes des ontologies. Par exemple, dans le monde XML, plusieurs approches traitent du problème des réponses à une requête en exploitant des documents de structure inconnue. Ainsi, [Näppilä *et al.*, 2008] ont défini une approche pour détecter des relations sémantiques dans des documents XML dont les noms des nœuds sont définis dans une DTD mais qui sont hétérogènes d'un point de vue structure. Leur méthode se base sur un calcul préalable des plus profonds graphes connexes dans l'arbre de document pour la requête posée. Ainsi, si l'utilisateur cherche un livre dont l'auteur est "T.B. Lee" et l'année de publication est "2001", le système recherche les graphes de nœuds (auteur, livre, année de publication) les plus profonds dans l'arbre XML du document. Parmi ces graphes, ceux dont les nœuds sont les plus proches deux à deux sont sélectionnés comme réponse. Bien que les résultats soient des ensembles de nœuds XML, le processus employé peut être vu comme une recherche de relations sémantiques entre les différents nœuds d'une réponse sans que le nom de la relation ne soit explicité.

2.3.3 Exploitation des connaissances de domaine

Les connaissances de domaine sont de plus en plus utilisées pour guider le processus de découverte de relations et améliorer sa précision. Nous distinguons plus particulièrement deux catégories de connaissances du domaine : (i) les instances de relations et (ii) les propriétés des relations (e.g. transitivité, fonctionnalité, intervalles de valeurs) ou toute autre règle d'inférence spécifique au domaine. Par

1. http://sprout.dfki.de/

exemple, si le processus de découverte d'instances de relations retrouve que la date de naissance d'une personne est le 01-01-2000 alors que la base de connaissances précise que cette même personne est décédée le 01-01-1999, l'instance de relation découverte ne peut pas être retenue.

• Comme nous l'avons vu, les instances de relations peuvent permettre d'apprendre des patrons d'extraction en tant qu'exemples (e.g. Snowball, KnowItAll, BOA). Elles sont également utilisées dans les processus de raisonnement exploitant des règles d'inférence.

• Dans un autre exemple, [Hignette *et al.*, 2009] utilisent les signatures des relations, les unités de mesures et les intervalles de valeurs pour découvrir les instances de concepts et de relations décrites dans des tables HTML.

Nous illustrons les travaux qui exploitent les instances de la base de connaissances, les propriétés des relations et des règles spécifiques au domaine à l'aide du système SOFIE. Ces travaux sont les plus proches à notre connaissance des travaux d'enrichissement de bases de connaissances effectués dans cette thèse ([Suchanek *et al.*, 2009]).

• *SOFIE* est un exemple de système qui vise à peupler une ontologie avec des instances de relations en exploitant des documents. Cette approche définit un ensemble de règles logiques permettant de vérifier que les faits extraits sont consistants avec les faits de la base de connaissances. Ainsi, ces règles vérifient qu'il y a adéquation entre les types des instances de concepts d'une instance de relation candidate et les domaines et co-domaines de cette même relation dans l'ontologie. L'approche proposée utilise aussi la *fonctionnalité* (inverse) des propriétés. Une relation R est dite fonctionnelle si une instance du concept source (domaine de R) peut être liée à au plus une instance du concept cible (co-domaine de R). Ainsi, si la relation *Lieu de Naissance* est déclarée comme fonctionnelle, toute personne ne pourra être reliée qu'avec un seul lieu de naissance.

SOFIE commence par appliquer des patrons linguistiques afin d'obtenir un ensemble d'instances de relations dites candidates. Le problème est ensuite vu comme un problème de satisfiabilité pondéré (Weighted Max-Sat) où les faits de la base de connaissances, les hypothèses/variables (faits candidats) et les règles sont mises sous forme clausale. Il s'agit alors de trouver une affectation des variables du problème qui maximise la somme des poids des clauses satisfaites. Cette affectation permet de déterminer quels faits candidats devraient être retenus.

Les expérimentations ont été menées en utilisant la base de connaissances Yago créée à partir des infobox et des catégories de Wikipédia. Des tests ont été effectués sur plusieurs corpus dont un corpus Wikipedia de 2000 documents. Les résultats obtenus montrent une bonne précision des faits extraits (précision supérieure à

80% pour 13 relations ciblées). L'utilisation de la base de connaissances a permis de filtrer les instances de relations candidates avec des règles logiques, ce qui a permis d'augmenter la précision des instances de relations retenues.

Cependant, comme le système SOFIE exploite des patrons lexicaux pour obtenir des instances de relations candidates, l'approche reste confrontée à la problématique de base qui est de devoir retrouver des régularités lexicales spécifiques dans les documents. Cette contrainte est bien sûr allégée si le nombre et la couverture des patrons augmentent mais reste tout de même restrictive. D'un autre côté, la majorité des relations recherchées n'ont pas de domaine/co-domaine communs (i.e. actedIn, bornIn, directed, establishedOnDate, hasArea, hasDuration, hasPopulation, hasProductionLanguage, hasWonPrize, locatedIn, writtenInYear) ce qui facilite le choix de la relation. Une exception est à noter pour les relations *actedIn* et *directed* définies entre les même concepts *Person* et *Film*, ce qui a conduit les auteurs à définir une règle de domaine discutable exprimant qu'une personne ne peut pas être à la fois acteur et directeur d'un film.

2.3.4 Conclusion

La découverte d'instances de relations sémantiques passe par la définition de patrons lexico-syntaxiques, par l'exploitation de parties de document ayant une structure régulière, ou par l'exploitation de bases de connaissances. La définition de patrons est complexe du fait de la nécessité d'en définir un grand nombre pour être le plus exhaustif possible. Des techniques d'apprentissage automatique, certes plus robustes, peuvent être appliquées mais elles requièrent un grand nombre d'exemples d'entrainement avec une bonne couverture et doivent être adaptées si elles sont appliquées à des corpus ayant des caractéristiques différentes. Par ailleurs, nous ne disposons pas toujours de parties de documents avec une structure régulière. Enfin ces approches ne permette pas d'exploiter les parties de document qui ne comportent pas de régularités lexico-syntaxiques ou structurelles.

Enrichir les annotations de document permet d'améliorer la recherche sémantique d'information en ajoutant des réponses (i.e. instances de concepts et/ou de relations). Une autre direction de recherche envisageable consiste à modifier les requêtes des utilisateurs. Il s'agit alors de rechercher des substituts à la relation présente dans la requête utilisateur en utilisant des heuristiques d'approximation ou de relaxation. L'étude de cette direction de recherche fait l'objet de la prochaine section.

2.4 Reformulation de requêtes sémantiques

Dans cette section, nous présentons deux catégories d'approches de reformulation de requêtes. Une première catégorie d'approches qui s'intéresse à approximer ou relaxer les requêtes sémantiques et une seconde catégorie d'approches qui combinent la recherche mots-clés et la recherche sémantique.

2.4.1 Approches d'approximation et de relaxation de requêtes

Plusieurs travaux sur la recherche sémantique de l'information (e.g. [Corby *et al.*, 2006; Hurtado *et al.*, 2006]) approximent les requêtes utilisateur en utilisant une ontologie de domaine.

• CORESE [Corby *et al.*, 2006] est un moteur de recherche sémantique basé sur les graphes conceptuels. Il permet l'interrogation SPARQL de données RDF(S) ou OWL-Lite et propose une approche d'approximation à la demande. L'utilisateur spécifie la nature de l'approximation à l'aide de prédicats ou de mots-clés insérés dans sa requête SPARQL [1]. Ces approximations consistent à (i) utiliser des concepts proches de ceux demandés par l'utilisateur (ii) rechercher des chemins de relations entre deux instances. L'utilisateur peut employer des approximations de diverses natures dans une même requête. Les réponses sont triées suivant leur proximité sémantique globale avec la requête initiale.

La proximité sémantique entre deux concepts est déduite du calcul de la distance sémantique entre toute paire de concepts de l'ontologie. Cette distance tient compte du fait que deux concepts à égale distance dans la hiérarchie de concepts sont considérés plus proches sémantiquement s'ils sont plus éloignés de la racine. Les réponses peuvent aussi être étendues à des instances d'autres concepts ou de relations quand ceux-ci sont liés aux concepts et relations de la requête par la propriété rdfs:seeAlso (si elle est déclarée dans l'ontologie).

D'un autre côté, CORESE permet aux utilisateurs de rechercher l'existence d'un chemin de relations entre deux ressources RDF longueur inférieure ou égale à un seuil fixé dans la requête. Ainsi, la requête { ?x *all* :: *parenteDirectAvec*{3} ?y} recherche les personnes X et Y qui ont un lien de parenté ne dépassant pas trois liens de parenté directs, alors que la requête { ?x *parenteDirectAvec*{3} ?y}

1. Le mot clé *more* dans l'entête de la requête SPARQL permet de déclencher les approximations

recherche les personnes X et Y qui ont un lien de parenté ne dépassant pas trois liens de parenté directs.

Le but principal de ces approximations est de permettre de répondre aux requêtes dans le cas d'absence des instances de concepts ou de relations recherchées. Quand des instances de concepts sont absentes, le système permet de retourner des instances de concepts proches sémantiquement. Par exemple, l'utilisateur pourra accéder aux instances du concept *Concert* même si elles ont été annotées de façon imprécise en tant qu'instances du concept *Événement*. Quand des instances de relations sont absentes, le système permet de retourner des instances de relations proches sémantiquement (avec le rdfs:seeAlso) ou donne à l'utilisateur la possibilité de formuler sa requête autrement avec les chemins de relations.

• L'approche proposée par [Hurtado *et al.*, 2006] transforme la recherche d'instances d'une relation sémantique, exprimée dans une requête SPARQL, en une recherche d'instances des classes correspondant au domaine et au co-domaine de cette relation. Cette méthode de relaxation vise en premier lieu à pallier l'incomplétude potentielle des bases de connaissances en supprimant la contrainte de liaison des instances tout en gardant un certaine cohérence sémantique par l'expression de jointures dans le corps de la requête.

Une autre direction de reformulation est aussi envisageable en exploitant les documents annotés pour transformer certaines contraintes sémantiques en des contraintes sur le document.

2.4.2 Approches combinant la recherche mots-clés et la recherche sémantique

D'autres approches, dites hybrides, s'attaquent à l'incomplétude des annotations sémantiques en combinant recherche sémantique et recherche mots-clés.

• Dans [Bhagdev *et al.*, 2008], la requête utilisateur est constituée de deux parties indépendantes : une partie sémantique correspondant à une requête SPARQL et une partie mots-clés. Par exemple, il est possible de poser une requête qui permet de rechercher tous les documents qui réfère à une instance de *ComposantElectronique* dont le texte contient la chaîne de caractères "Résistance". Les résultats de la requête utilisateur considérée dans sa globalité sont les documents obtenus à la fois par les parties mots-clés et sémantiques de la requête. En cas d'absence de réponses à l'une ou l'autre des parties de la requête, les réponses correspondent à la partie mots-clés seule ou à la partie sémantique uniquement.

• Dans l'approche proposée par [Castells *et al.*, 2007], l'utilisateur pose des re-

quêtes sémantiques en RDQL [1] et les documents retournés sont triés (i) suivant leur degré de similarité sémantique avec la requête et (ii) suivant leur degré de similarité lexicale avec les mots-clés extraits de la requête. L'extraction des mots-clés de la requête consiste à extraire les labels des concepts, propriétés et instances cités dans la requête. La similarité document-requête tient compte aussi du degré de confiance associé aux annotations des documents d'une part et de l'importance des variables de la requête d'autre part. Cette importance peut être fixée par l'utilisateur ou par le système. Une mesure de similarité globale est calculée par combinaison des valeurs de similarité lexicale et sémantique. L'utilisateur peut aussi définir un coefficient à associer à chacune des similarités lexicale et sémantique afin de calculer la similarité globale en conséquent.

2.4.3 Synthèse et conclusion

La reformulation des requêtes utilisateur au moment de l'interrogation peut très facilement générer des réponses non pertinentes pour l'utilisateur. Plusieurs approches de reformulation exploitent l'ontologie de domaine pour obtenir plus de réponses en relaxant ou en approximant des contraintes exprimées dans la requête utilisateur. Dans CORESE, les instances de relations sont soient approximées par un *rdfs*:*seeAlso*, soient étendues par la recherche de chemins de relation. Dans [Hurtado *et al.*, 2006] les instances de relations recherchées sont supprimées progressivement de la requête.

Les approches de recherche hybrides permettent de combiner la recherche sémantique et la recherche mots-clés. Il s'agit d'un moyen pour pallier l'incomplétude des annotations sémantiques réalisées sur les documents et d'augmenter le rappel. Cependant, ces approches éliminent potentiellement certaines contraintes sémantiques de la requête. Par ailleurs, l'utilisation des mots-clés réinjecte les problèmes liés à la synonymie et à l'ambiguïté dans la recherche d'information. Par exemple, dans [Castells *et al.*, 2007] les mots-clés sont des labels de concepts, propriétés et instances. Ils ne permettent pas forcément d'accéder à tous les documents pertinents. Par ailleurs, la fixation manuelle des coefficients utilisés pour combiner les similarités lexicales et sémantiques est complexe. Les valeurs de coefficient à considérer ne sont pas forcément les mêmes d'un corpus à un autre, elles dépendent aussi des annotations associées aux documents traités.

Ce type d'approches pose également le problème de la combinaison de deux méthodes de recherche : quand privilégier l'une ou l'autre de ces recherches ? Comment combiner les similarités calculées pour chaque type de recherche et obtenir

1. http://www.w3.org/Submission/RDQL/

des documents à retourner triés par ordre de pertinence décroissante ?

Dans la prochaine section, nous positionnons nos travaux par rapport à ceux présentés dans cet état de l'art.

2.5 Positionnement

L'état de l'art que nous venons de présenter a porté d'une part sur l'annotation sémantique des documents semi-structurés et d'autre part sur la reformulation de requêtes sémantiques. Ces deux types d'approches permettent d'augmenter le nombre de réponses aux requêtes exprimées dans le vocabulaire d'une ontologie de domaine. Les annotations auxquelles nous nous sommes intéressées permettent d'annoter des entités de document avec (i) des concepts, (ii) des instances de concepts ou (iii) des instances de relations.

Notre travail sur l'enrichissement et la reformulation a été effectué dans l'objectif d'exploiter les documents semi-structurés pour retrouver des réponses (nouvelles) aux requêtes utilisateur recherchant des instances de concept et de relations, tout en contrôlant leur pertinence. Nous nous sommes également fixés la contrainte de ne pas recourir à des régularités lexico-syntaxiques ou structurelles dans l'exploitation de ces documents. Cette contrainte permet de découvrir des connaissances qui ne peuvent pas être proposées par les approches que nous avons exposées dans cet état de l'art.

Nous avons vu que les approches de découverte d'instances de concepts dans les documents obtiennent de bons résultats, en particulier pour les entités nommés. Aussi, une approche recherchant les instances de relation dans les documents peut se baser sur ces résultats. En revanche, le problème de recherche d'instances de relations sémantiques entre les instances de concepts utilisées pour l'annotation est plus difficile à résoudre, notamment si les documents ne contiennent pas de régularités lexico-syntaxiques ou structurelles exploitables.

Nous avons d'abord travaillé sur la reformulation de requêtes sémantiques. Plusieurs des approches de reformulation présentées dans cet état de l'art exploitent uniquement l'ontologie pour approximer certaines contraintes sans exploiter le contexte des instances dans les documents. Ces approches ne sont pas adaptées pour retrouver des réponses si les instances de concept ne sont pas identifiées (cas des annotations par concepts). D'autres approches introduisent des mots-clés pour effectuer une recherche hybride, ce qui supprime des contraintes sémantiques de la requête.

Dans notre travail, nous proposons de répondre à des requêtes portant sur des

instances de concepts et de relations, dans le cas ou les documents semi-structurés sont annotés par des concepts uniquement, tout en tenant compte des entités de document annotées et de leur structuration dans les documents.

Ainsi, dans une première approche, appelée *SHIRI-Querying*, nous nous sommes fixés comme objectif d'obtenir des réponses aux requêtes sémantiques en retournant des parties de documents correspondant à des nœuds de leur arbre DOM. Cette transformation du problème de la recherche d'instances de concepts et/ou de relations vers la recherche de parties de document référant à ces instances offre une solution pour répondre aux requêtes sémantiques dans le cas ou les documents ont été annotés par les concepts uniquement et où les instances ne sont pas identifiées.

Notre approche consiste à transformer les contraintes sur des instances de concept et de relations en contraintes sur des nœuds annotés par des concepts du domaine. Les heuristiques employées lors de cette transformation se basent sur la description des concepts et des relations dans une ontologie de domaine mais également sur des informations concernant la composition des nœuds annotés en termes de concepts et leur voisinage structurel dans les documents.

Dans une deuxième partie de cette thèse, nous avons travaillé sur l'enrichissement de bases de connaissances à partir de documents semi-structurés annotés par des instances de concepts. Cet enrichissement consiste à produire de nouvelles instances de relations entre les instances de concepts utilisées pour l'annotation.

De nombreuses approches de découverte d'instances de relations sémantiques se fondent souvent sur l'existence de régularités lexico-syntaxiques ou structurelles dans les documents et sont inopérantes lorsque ces régularités n'existent pas. Dans notre travail, nous nous sommes intéressés à ce problème d'extraction d'instances de relations dans un contexte où les documents n'ont pas de telles régularités. Ceci nous a conduit à proposer une approche différente, *REISA*, qui permet de produire des instances de relations sémantiques à partir de documents hétérogènes.

Les instances de relations générées par notre approche sont d'abord sélectionnées par le simple voisinage structurel des références d'instances de concept dans le document. Elles sont ensuite contrôlées et filtrées respectivement par (i) les axiomes de fonctionnalité OWL (inverse) des propriétés, (ii) des règles spécifiques au domaine d'application. Ces contrôles sont similaires à ceux effectués par l'approche SOFIE [Suchanek *et al.*, 2009], cependant, nous n'utilisons pas de patrons lexico-syntaxiques pour obtenir les instances de relations candidates.

Nous proposons un contrôle supplémentaire effectué via le Web, inspiré par des approches telles que CPankow [Cimiano *et al.*, 2005] et SHIRI-Extract [Thiam

et al., 2009], pour valider des expressions textuelles représentant des instances de relations candidates.

Les instances de relations retenues après ces différents contrôles sont pondérées automatiquement suivant la distance entre les entités de documents référant aux instances de concepts liées par la relation.

Avec un objectif similaire à [Buche *et al.*, 2008], nous avons aussi proposé une approche d'interrogation afin d'exploiter les mesures de confiance associées aux faits. Cette approche reformule des requêtes sémantiques afin (i) d'accéder d'une façon transparente aux bases de connaissances préexistantes ou construites par enrichissement et (ii) d'expliciter les poids des faits afin de trier les réponses finales.

Dans le prochain chapitre nous décrivons le modèle sémantique d'intégration *SIM* qui nous permet de formaliser de façon homogène les différents éléments entrant en jeu dans la problématique posée : entités de document, connaissances du domaine et liens d'annotation. Dans les chapitre 4 et 5 nous présentons l'approche *SHIRI-Querying* et son évaluation. Dans les chapitres 6 et 7 nous présentons l'approche *REISA* et son évaluation.

Chapitre 3

Modèle sémantique d'intégration

Plusieurs modélisations ont été proposées ou employées pour représenter les entités de document, les connaissances de domaine et leurs liens. Par exemple, DBpedia utilise l'attribut $foaf : page$, défini dans l'ontologie Foaf[1] pour lier des instances de concepts définis dans la base de connaissances DBpedia à des articles Wikipedia les décrivant. KIM utilise une représentation objet au format textuel structuré JSON[2] et indique l'instance associée à un terme par le champ "instance".

Pour intégrer des annotations et des connaissances provenant de différentes sources, nous avons besoin de modéliser les annotations et notamment les liens entre les connaissances et les entités de document de façon homogène quelque soit la provenance de ces informations ou l'outil utilisé pour les produire.

1. Ontologie FOAF (Friend Of A Friend), http://xmlns.com/foaf/spec/
2. JSON (JavaScript Object Notation), http://www.json.org/

Dans ce chapitre nous présentons dans un premier temps le modèle sémantique d'intégration *SIM* (Semantic Integration Model) que nous avons défini pour représenter les bases de connaissances, les entités de document et leurs liens dans un cadre homogène. Dans un deuxième temps, nous formalisons le problème étudié dans les termes du modèle *SIM* et des langages standards RDF et SPARQL.

3.1 Le modèle *SIM*

Dans cette section nous présentons d'abord par une description générale du modèle. Nous détaillons les deux principaux niveaux du modèle, à savoir, le niveau connaissances et le niveau annotation.

3.1.1 Description générale du modèle *SIM*

Le modèle sémantique d'intégration *SIM* a pour objectif de définir un niveau d'abstraction au dessus des ontologies de domaine afin de permettre l'intégration des bases de connaissances et des annotations de document et de modéliser la confiance accordée aux faits. Plus précisément, il permet une représentation homogène des entités de documents, des bases de connaissances et des liens entre les entités de documents et les connaissances de domaine auxquelles elles réfèrent. Les annotations de document sont séparées en deux parties dans le modèle *SIM* : (i) une partie représentant les entités de document annotées et (ii) une partie représentant les connaissances de domaine extraites ou construites par annotation et qui sont considérées comme étant des bases de connaissances à part entière, ayant une mesure de confiance spécifique.

Les liens représentés sont des liens entre des entités de document d'une part et des instances de concepts, des concepts et des types de littéraux d'autre part. Des concepts et propriétés sont aussi définis pour représenter numériquement la confiance accordée aux faits en regroupant les faits ayant un même poids dans le même graphe RDF nommé.

Le modèle SIM est représenté RDF(S)/OWL. La figure 3.1 présente les différents niveaux et éléments du modèle qui sont détaillés dans les sections suivantes.

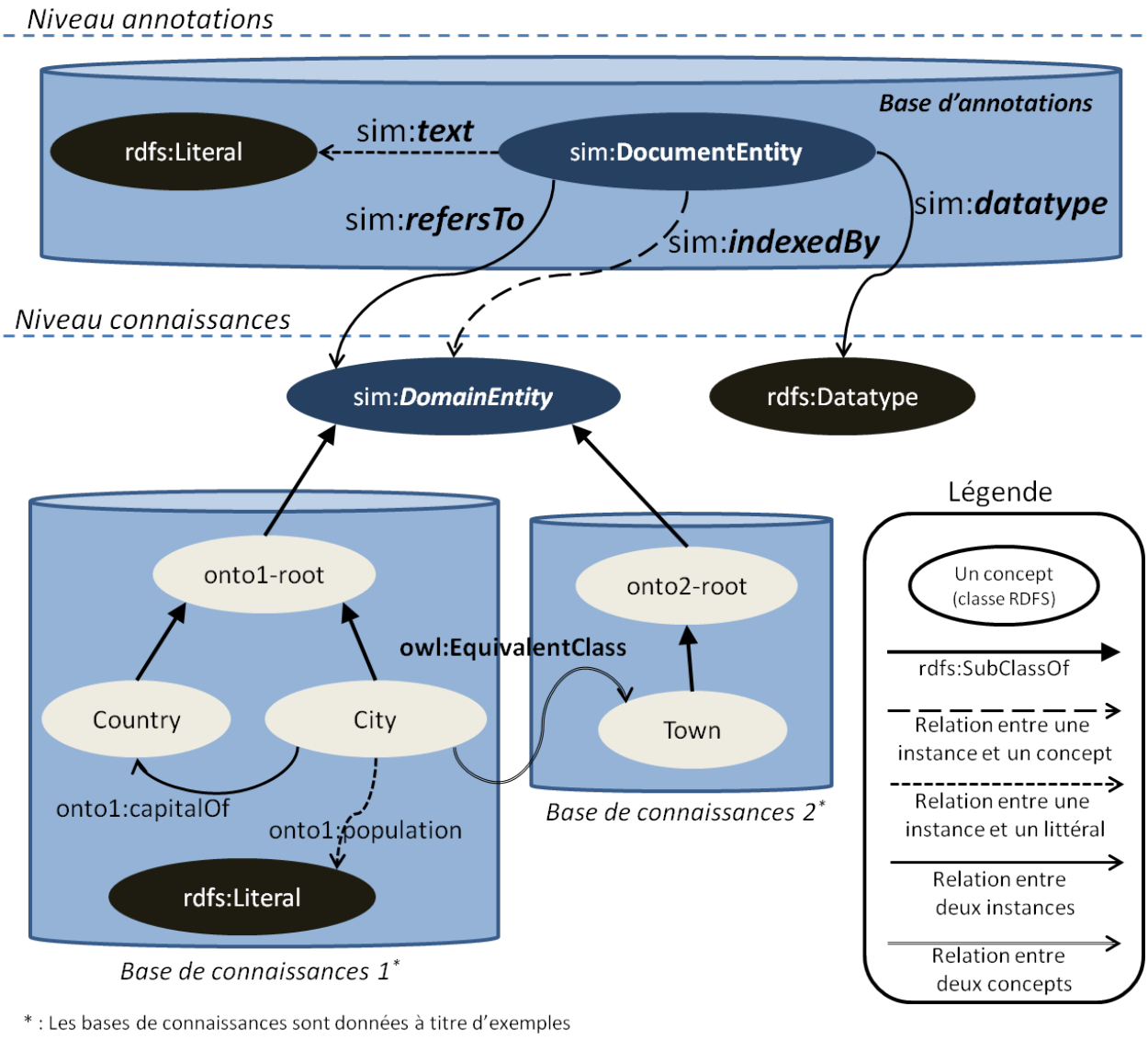

FIGURE 3.1 – Modèle Sémantique d'Intégration (SIM)

3.1.2 Niveau connaissances

Ce niveau permet d'intégrer toutes les bases de connaissances exploitées/produites par l'annotation, c'est à dire, les connaissances du domaine provenant des bases de connaissances RDF préexistantes ou des bases de connaissances issues de l'annotation.

Base de connaissances. Une base de connaissances est caractérisée par une ontologie de domaine et un ensemble de faits décrivant les instances de l'ontologie. Nous représentons les différents éléments d'une base de connaissances de la façon suivante, en donnant les équivalents OWL de chaque notation :

1. **Une ontologie de domaine** O^i est définie par le quintuplet (C_{O^i}, R_{O^i}, A_{O^i}, X_{O^i}) où :
   - C_{O^i} est l'ensemble des concepts (i.e. owl:Class).
   - R_{O^i} est l'ensemble des relations : propriétés dont le domaine et le co-domaine sont des concepts (i.e. owl:ObjectProperty)
   - A_{O^i} est l'ensemble des attributs : propriétés dont le domaine est un concept et le co-domaine est un littéral (i.e. owl:DatatypeProperty).
   - X_{O^i} est un ensemble d'axiomes définissant les caractéristiques des concepts et des propriétés et éventuellement des alignements avec d'autres ontologies tel que :
     - `domain(P,C)` indique que le domaine de la propriété P est C (i.e. <p rdfs:domain C>)
     - `range(P,C)` indique que le co-domaine de la propriété P est C (i.e. <p rdfs:range C>)
     - `subClass(`C_1`,`C_2`)` indique que C_1 est un sous-concept de C_2 (i.e. $< C_1$ rdfs:subClassOf $C_2 >$)
     - `subProperty(`P_1`,`P_2`)` indique que P_1 est une spécialisation de P_2 (i.e. $< P_1$ rdfs:subPropertyOf $P_2 >$).
     - `fn(`P`)` indique que la propriété P est fonctionnelle (i.e. $< P_1$ rdf:type owl:FunctionalProperty>).
     - `ifn(`P`)` indique que la propriété P est inverse fonctionnelle ($<$ P_1 rdf:type owl:InverseFunctionalProperty>).
     - `equivalentConcept(`C_1`,`C_2`)` indique que C_1 est équivalent à C_2 (i.e. $< C_1$ owl:EquivalentClass $C_2 >$)
     - `equivalentProperty(`P_1`,`P_2`)` indique que P_1 est équivalente à P_2 (i.e. $< C_1$ owl:EquivalentProperty $C_2 >$)
     - `sameAs(`i_1`,`i_2`)` indique que les instances i_1 et i_2 réfèrent à la même entité du monde réel bien qu'ayant deux identifiants (IRI) différents.
2. **Un ensemble de faits, noté** $T^j_{O^i}$**,** décrivant les instances de O^i.

Les figures 3.2 et 3.3 présentent deux extraits des faits décrivant les instances de la base de connaissances de WKB de KIM et de la base de connaissances $DBpedia$ (faits RDF représentés en notation $N-TRIPLE$).

wkb:Laos.0 **rdf:type** proton:Country
wkb:Laos.0 **proton:partOf** wkb:Continent.2
wkb:Continent.2 **rdf:type** proton:Continent
wkb:Continent.2 **rdfs:label** "Asia"
wkb:Vientiane.0 **rdf:type** proton:City
wkb:Vientiane.0 **proton:capital** wkb:Laos.0

FIGURE 3.2 – Extrait des descriptions d'instances (T_O) de la base WKB de KIM

<dbpedia:Pyrenees> <rdf:type> <dbpedia:ontology/MountainRange>
<dbpedia:Pyrenees> <rdf:type> <dbpedia:class/yago/MountainRangesOfFrance>

FIGURE 3.3 – Extrait des descriptions d'instances (T_O) de *DBpedia*

Le concept *DomainEntity* est une abstraction de tous les concepts décrits dans les ontologies de domaine couvertes par les bases de connaissances considérées. Tout concept de ces ontologies est une sous-classe de *DomainEntity*. Ainsi les instances des concepts du domaine sont aussi des instances du concept *DomainEntity*. Cette abstraction permet de définir le lien entre les entités de document et les instances de concepts du domaine quelque soit la base de connaissances considérée.

Dans la figure 3.1, deux bases de connaissances sont représentées à titre d'exemple, mais le nombre de bases de connaissances représentables n'est pas limité. Dans cet exemple, deux ontologies regroupant des concepts (e.g. Conference, Topic) et des propriétés RDF (e.g. hasTopic) du domaine des évènements scientifiques sont alignées avec des liens d'équivalence entre classes (e.g. onto1 :City est déclaré équivalent à onto2 :Town).

Les types de données littérales sont aussi représentés avec la classe *rdfs:datatype*. Les instances de la classe *rdfs:datatype* sont les types de données XML Schema tels que les dates, les années ou les nombres entiers. Aussi, cette représentation exprime le fait qu'une entité de document contient un littéral ayant un type spécifique.

3.1.3 Niveau annotation

Le niveau annotation permet de représenter les entités de documents annotées (*sim:DocumentEntity*), leur contenu textuel (*sim:text*) et leurs liens avec les instances de concept du domaine (*sim:refersTo*), avec les types de données (*sim:datatype*)

ou avec des concepts du domaine (*sim:indexedBy*). Toutes ces données sont regroupées dans ce que nous appelons ***Base d'annotations***.

La figure 3.4 présente un extrait de base d'annotations mis en conformité avec le modèle SIM.

corpus:doc0/html/body/div/p[3]/a.0 **rdf:type** sim :DocumentEntity
corpus:doc0/html/body/div/p[3]/a.0 **sim:refersTo** kimkb:Vietnam.0
corpus:doc0/html/body/div/p[3]/a.0 **sim:text** "Vietnam"
corpus:doc0/html/body/div/p[3].12 **sim:refersTo** kimkb:Laos.0
corpus:doc0/html/body/div/p[3].12 **sim:text** "Laos"
corpus:doc0/html/body/div/p[3].20 **sim:refersTo** kimkb:Mekong.0
corpus:doc0/html/body/div/p[3].20 **sim:text** "Mékong"
corpus:doc0/html/body/div/p[3]/a[2].0 **sim:refersTo** kimkb:Hanoi.0
corpus:doc0/html/body/div/p[3]/a[2].0 **sim:text** "Hanoi"
corpus:doc0/html/body/div/p[3].35 **sim:datatype xsd:Year**
corpus:doc0/html/body/div/p[3].35 **sim:text** "1945"
corpus:doc0/html/body/div **sim:indexedBy** *onto:Country*
corpus:doc0/html/body/div/p[3] **sim:indexedBy** *onto:City*

FIGURE 3.4 – Exemple d'annotations conformes au modèle SIM

Les entités de document sont des instances du concept *DocumentEntity*. Chaque entité de document représente une partie de document, annotée sémantiquement par au moins une instance de concept, un type de littéral ou un concept du domaine. Il peut s'agir d'un document entier, d'un nœud de l'arbre DOM du document ou d'une séquence de mots. Chaque entité de document est identifiée par une URI. Par exemple, dans la figure 3.4, les URI des entités de document sont constituées de la concaténation de l'URL de document, du chemin $XPATH$ du nœud de document contenant l'entité et potentiellement de la position du premier caractère de l'entité dans le nœud si l'entité est un terme.

L'attribut *sim:text* modélise le contenu textuel d'une entité de document donnée, représentée sous la forme d'un littéral. Ses valeurs dépendent du type de l'entité de document, elles correspondent :
- au contenu textuel entier du document si l'entité est un document.
- au texte constitué par l'ensemble des mots si l'entité est une séquence de mots quelconque.
- au texte constitué par la concaténation des nœuds textuels qui sont fils directs de l'entité dans l'arbre DOM, si l'entité est un nœud du document considéré.

Exemple 1. Dans la figure 3.4, le triplet <corpus:doc0/html/body/div/p[3].12> *sim:text* "Laos" exprime le fait que l'entité de document dont l'IRI est *<corpus:doc0/html/body/div/p[3].12>* a pour contenu textuel le terme "Laos".

La relation *sim*:*refersTo* lie une entité de document e à une instance de concept i, notée *refersTo(e,i)*). Elle indique que e mentionne (ou réfère à) i, instance de concept.

Exemple 2. Dans la figure 3.4, le triplet <corpus:doc0/html/body/div/p[3].12> *sim*:*refersTo* kimkb:Laos.0 exprime le fait que l'entité de document dont l'URI est <`corpus:doc0/html/body/div/p[3].12`> est une représentation textuelle de l'instance de KIM *kimkb*:*Laos.0*.

La relation *sim*:*datatype* lie des entités de document à des types de données XMLSchema (instances de *rdfs*:*datatype*) tels que *xsd*:*Date* ou *xsd*:*Double*. Elle indique que le contenu d'une entité de document correspond à une valeur possible du type de données associé.

Exemple 3. Dans la figure 3.4, le triplet <corpus:doc0/html/body/div/p[3].35> *sim*:*datatype* xsd:Year exprime le fait que l'entité de document dont l'URI est <`corpus:doc0/html/body/div/p[3].35`> correspond à une valeur possible d'année.

La relation *sim*:*indexedBy* lie des entités de document à des concepts du domaine. Elle indique qu'une instance du concept a été localisée dans une entité de document sans être répertoriée dans une base de connaissances (i.e. instance sans identifiant et donc sans aucune description).

Exemple 4. Dans la figure 3.4, le triplet corpus:doc0/html/body/div/p[3] *sim*:*indexedBy* onto:City exprime le fait que l'entité de document dont l'URI est `corpus:doc0/html/body/div/p[3]` contient une instance de Ville (i.e. *onto*:*City*) non identifiée et est donc indexée par le concept *onto*:*City*

3.2 Formalisation de la problématique étudiée

Dans cette section, nous présentons une formalisation de la problématique étudiée en utilisant le modèle SIM. Étant donné que les requêtes utilisateurs que nous traitons sont formulées en SPARQL suivant une ontologie de domaine, nous introduisons dans un premier temps les notions RDF et SPARQL nécessaires à la définition de ces requêtes, que nous appelons *requêtes sémantiques*. Dans un

second temps nous présentons les contextes dans lesquels ce problème d'interrogation est étudié.

3.2.1 Définitions préliminaires

Nous introduisons ici successivement les notions de termes RDF, de graphes RDF, de *dataset* RDF, de patrons de triplets RDF et de patrons de graphe RDF élémentaires (ou basiques) tels que définis dans la spécification SPARQL [1] du W3C [2].

Terme RDF [3]**.** Considérons les ensembles infinis deux à deux disjoints I, $RDF\text{-}B$, et $RDF\text{-}L$ (IRIs, nœuds Blancs et Littéraux). L'ensemble des termes RDF est $RDF\text{-}T = \mathrm{I} \cup \mathrm{RDF}-\mathrm{L} \cup \mathrm{RDF}-\mathrm{B}$.

Graphe RDF. Un graphe RDF est un ensemble de triplets $(s, p, o) \in RDF\text{-}T \times I \times RDF\text{-}T$. La définition inclue la possibilité de sujets littéraux, cependant, les syntaxes proposées actuellement ne le prennent pas en compte et la définition pratique d'un patron de triplet revient à $(s, p, o) \in (I\ cup\ RDF\text{-}B) \times I \times RDF\text{-}T$.

***Dataset* RDF** [4]**.** Un dataset RDF est un ensemble : { G, $(< u_1 >, G_1)$, $(< u_2 >, G_2)$, . . . $(< u_n >, G_n)$ } où G et chaque graphe RDF G_i sont des graphes et chaque $< u_i >$ est une IRI. Chaque $< u_i >$ est distinct. G est appelé graphe par défaut. Les $(< u_i >, G_i)$ sont appelés des **graphes RDF nommés**.

Variable de requête [5]**.** Une variable de requête appartient à l'ensemble V où V est infini et disjoint de $RDF\text{-}T$.

Patron de triplet RDF [6]**.** Un patron de triplet est un triplet $(s, p, o) \in (RDF\text{-}T \cup V) \times (I \cup V) \times (RDF\text{-}T \cup V)$. La définition inclut la possibilité de sujets littéraux, cependant, les syntaxes proposées actuellement ne le prennent pas en compte et la définition pratique d'un patron de triplet revient à $(s, p, o) \in (I \cup V) \times (I \cup V) \times (RDF\text{-}T \cup V)$

Patron de graphe RDF élémentaire [7]**.** Un patron de graphe élémentaire est un ensemble de patrons de triplets.

1. http://www.w3.org/TR/rdf-sparql-query/#sparqlDefinition
2. http://www.w3.org
3. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/#section-data-model
4. http://www.w3.org/TR/rdf-sparql-query/#sparqlDataset
5. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/#section-data-model
6. http://www.w3.org/TR/rdf-sparql-query/#sparqlBasicTerms
7. http://www.w3.org/TR/rdf-sparql-query/#BasicGraphPatterns

Dans le cadre de cette thèse, nous considérons un sous-ensemble de requêtes SPARQL [1] de la forme *SELECT* constituées de blocs de triplets représentant un graphe RDF élémentaire et de contraintes sur les variables du patron de graphe représentées par des clauses *FILTER*. À un niveau domaine, les requêtes ciblées recherchent des instances de concepts du domaine, vérifiant potentiellement des relations sémantiques avec d'autres instances, ou des valeurs d'attributs littéraux. Dans un souci de simplification, nous définissons ces requêtes, que nous appelons *requêtes sémantiques*, comme suit :

Requête sémantique. Une requête sémantique q formulée suivant une ontologie de domaine Ω est définie par le quadruplet (P, S, F, D) où :

- P est un patron de graphe basique conforme à Ω. $V(P)$ dénote l'ensemble des variables qui sont utilisées dans P et $C(P)$ dénote l'ensemble des concepts utilisés dans la requête pour typer les instances recherchées.
- F est un ensemble de contraintes définies par la combinaison logique d'expressions booléennes e. $e_v \in F$ dénote une expression utilisant la variable v.
- S est l'ensemble des variables sélectionnées (apparaissant dans la clause SELECT de la requête).
- D est le *dataset* RDF interrogé. La réponse à la requête se fera en alignant P et F avec D.

3.2.2 Contextes d'étude

Le problème que nous étudions, fournir des réponses (nouvelles) à la recherche sémantique de l'information, se fait dans deux cas de figure différents. Un premier cas de figure où les documents semi-structurés sont annotés avec des concepts du domaine (cf. figure 3.5) et un deuxième cas de figure où les documents sont annotés par des instances de concepts (cf. figure 3.6). Dans les deux cas, les requêtes posées sont des requêtes sémantiques telles que définies dans la section 3.2.1.

Dans le premier cas, des outils externes sont utilisés pour annoter un corpus cible par des concepts définis dans une ontologie de domaine. Nous représentons ensuite ces annotations dans les termes du modèle *SIM*. La base d'annotations représente les entités de document annotées comme étant des instances du concepts *DocumentEntity*. Les annotations sont représentées par des liens *sim:indexedBy* entre les instances de *DocumentEntity* créées et des concepts du domaine. Nous ne disposons pas dans ce premier cas d'une base de faits décrivant les instances des concepts.

1. http://www.w3.org/TR/rdf-sparql-query/#sparqlQuery

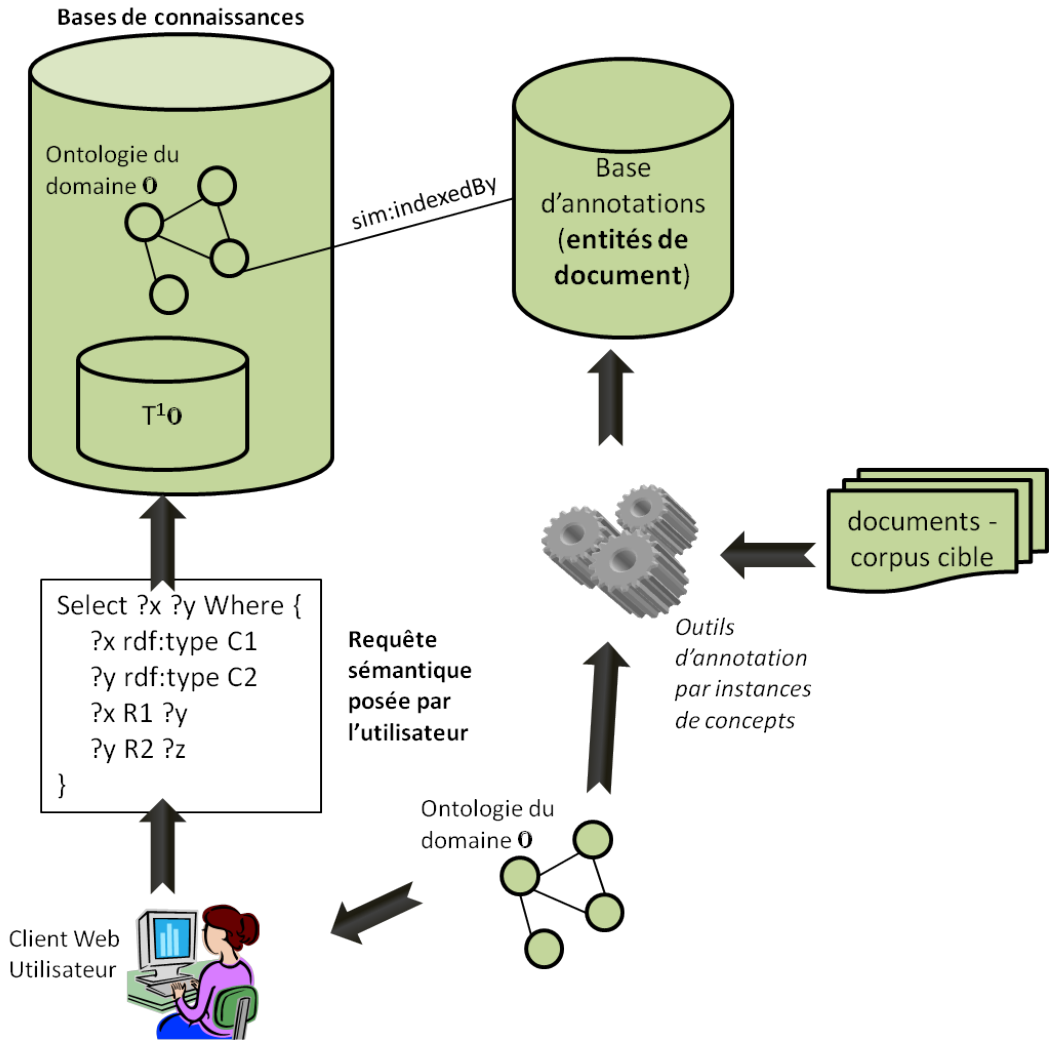

FIGURE 3.5 – Premier contexte d'étude : Annotation par concepts

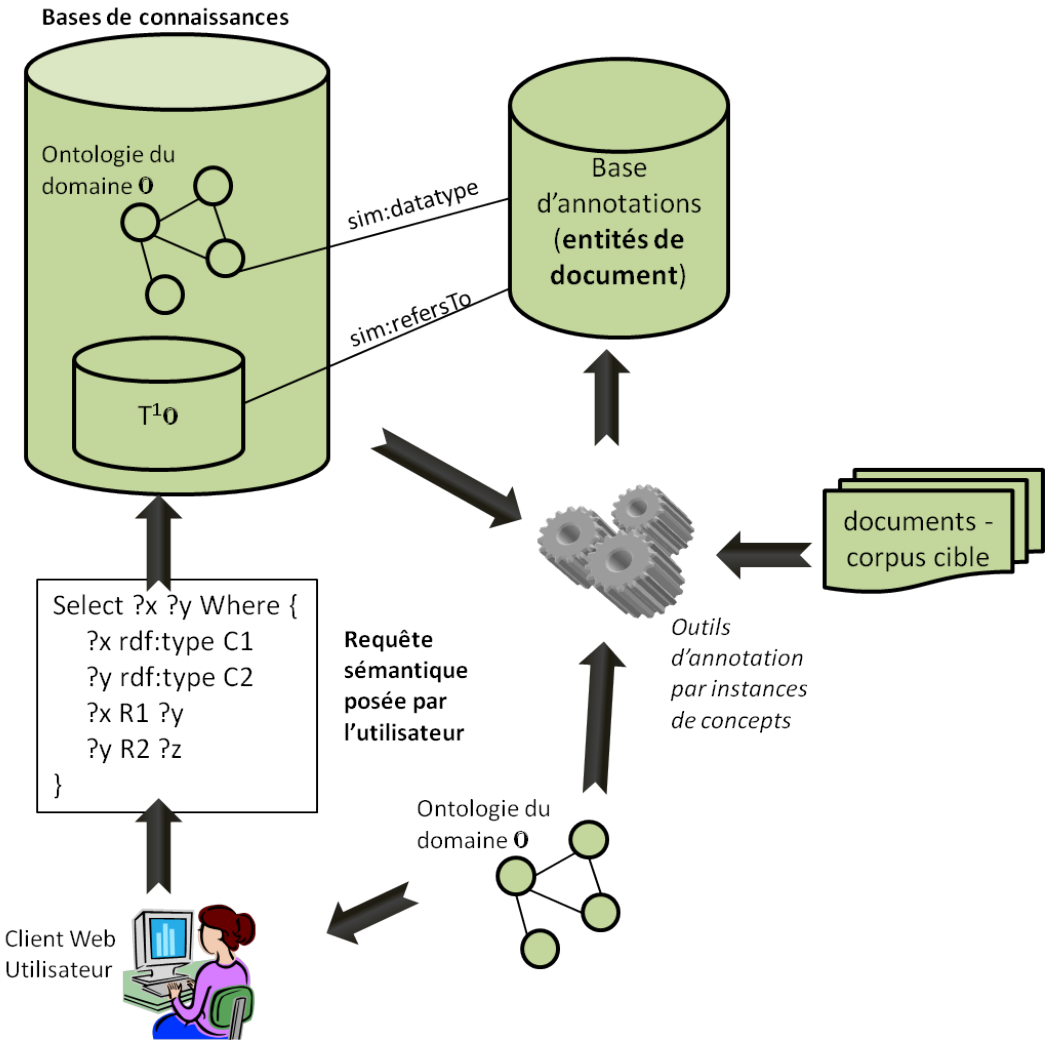

FIGURE 3.6 – Deuxième contexte d'étude : Annotation par instances de concepts

Dans le deuxième cas, des outils externes (ou utilitaires de notre développement) sont utilisés pour annoter un corpus cible par des instances de concepts définis dans une base de connaissances préexistantes ou crées à la volée lors de l'annotation. Comme dans le premier cas, nous représentons ces annotations dans les termes du modèle *SIM*. La base d'annotations représente aussi les entités de document annotées comme étant des instances du concept *DocumentEntity*. Les annotations sont représentées par des liens *sim:refersTo* entre les instances de *DocumentEntity* créées et des instances des concepts du domaine. La propriété *sim:datatype* représente les types de littéraux associés aux entités de document par les différents annotateurs.

Ainsi, dans ce cas, les bases de connaissances interrogées peuvent être issues de l'annotation ou préexistantes. Notons ici qu'une seule ontologie de domaine est représentée, cependant, dans la pratique les différentes bases de connaissances peuvent employer des ontologies différentes $O_1, O_2, .., O_n$. Dans ce cas l'ontologie O dans la figure 3.6 correspond à l'union de toutes ces ontologies et les bases de faits $T_O^1, T_O^2, .., T_O^n$ de chaque base de connaissances sont saturées suivant les axiomes et les alignements de ces différentes ontologies.

Dans la première partie de cette thèse, nous nous intéressons au cas où les documents sont annotés par des concepts du domaine. Dans la deuxième partie, nous nous intéressons au cas où les documents sont annotés par des instances de concept répertoriées dans des bases de connaissances les décrivant.

3.3 Conclusion

Dans ce chapitre nous avons présenté le modèle sémantique d'intégration *SIM* qui permet de représenter d'une façon homogène les entités de document annotées, les connaissances du domaine et les liens d'annotation. Nous avons formalisé le type de requêtes auquel nous nous intéressons, requêtes que nous avons appelées « requêtes sémantiques ». Enfin, nous avons présenté les deux contextes d'étude qui vont être considérés dans la suite de ce manuscrit.

Dans le prochain chapitre, nous présentons l'approche *SHIRI-Querying* qui répond à la problématique d'interrogation sémantique de documents semi-structurés annotés par des concepts du domaine. Le chapitre 5 sera dédié à l'évaluation de cette approche.

Chapitre 4

Reformulation de requêtes pour l'interrogation sémantique de documents semi-structurés

4.1 Introduction

Problématique

La recherche sémantique de l'information est une des principales motivations du Web sémantique. Un moteur de recherche sémantique peut être vu comme un outil qui répond à des requêtes – formulées avec les concepts et les relations d'une ontologie de domaine – en les alignant avec des annotations sémantiques des documents cibles ou des connaissances préexistantes. Dans une vue idéale, ce problème de Recherche d'Information (RI) peut être considéré comme similaire à la RI dans les bases de données relationnelles où les réponses sont des ensembles de tuples satisfaisant la requête de l'utilisateur. Cependant, cette vue ne se concrétise que si les contenus de tous les documents peuvent être représentés par des instances de concepts ou de relations définis dans une ontologie donnée.

Les avancées de la recherche visant à automatiser le peuplement des ontologies et l'annotation des documents sont prometteuses [Cimiano *et al.*, 2005; Etzioni *et al.*, 2004; Popov *et al.*, 2004; Thiam *et al.*, 2009]. Cependant, la localisation précise de toutes les instances dans un document reste une tâche difficile. Cette précision dépend aussi de la granularité choisie : termes, nœuds de documents (e.g. balises XML ou HTML) ou documents entiers. Dans ce cadre, plusieurs annotateurs choisissent d'annoter les documents avec des concepts uniquement et ne prennent pas le risque d'identifier précisément les instances de concepts. D'un autre côté, d'autres annotateurs créent des instances de concepts à la volée mais ne leur associent aucune description à part leur concept (e.g. KIM [Popov *et al.*, 2004], Cpankow [Cimiano *et al.*, 2005]). Dans de tels cas, les requêtes sémantiques portant sur les instances de concept et de relations n'auront pas de réponses.

Nous proposons de répondre à cette problématique en utilisant des heuristiques permettant de retourner des parties de document référant aux instances de concepts et de relations recherchées, à partir de documents annotés par des concepts de l'ontologie de domaine.

Le projet SHIRI

SHIRI[1] est un système pour l'intégration de documents semi-structurés relatifs à un domaine d'application décrit par une ontologie.

Le but du système SHIRI est de permettre aux utilisateurs d'accéder à des parties de documents pertinentes en interrogeant les documents avec des requêtes sémantiques. Les langages W3C standard RDF(S)/OWL sont utilisés pour la re-

1. Système Hybride d'Intégration et de Recherche d'Information, Digiteo labs project (LRI, SUPELEC)

présentation et l'annotation des parties de document et le langage SPARQL est utilisé pour leur interrogation. L'approche SHIRI-Querying a été proposée dans ce cadre pour répondre au problème d'interrogation posé.

L'approche SHIRI-Querying

L'objectif de *SHIRI-Querying* est de répondre à des requêtes sémantiques (cf. chapitre 3) en retournant des nœuds de document pouvant contenir l'information recherchée, dans un contexte où les documents sont annotés par des concepts de l'ontologie de domaine.

Nous utilisons une annotation sémantique des nœuds proposée par [Thiam *et al.*, 2009] [1] pour reformuler des requêtes sémantiques de façon à rechercher des nœuds de documents et non plus des instances de concepts et de relations. Nous trions ces reformulations suivant des heuristiques que nous définissons pour caractériser leur pertinence par rapport à l'information recherchée par la requête initiale de l'utilisateur. Ces heuristiques ont été formalisées avec une relation d'ordre entre les requêtes reformulées, dans le but de trier les réponses finales suivant leur précision.

La figure 4.1 présente l'architecture générale de *SHIRI-Querying*. L'approche est composée de deux principaux modules. L'adaptateur qui prend en entrées des documents semi-structurés contenant des termes annotés par des concepts pour annoter les nœuds de document conformément au modèle *SHIRI-Annot*. Le *moteur de requêtes* qui applique notre approche de reformulation et de trie pour construire des requêtes reformulées à partir de la requête SPARQL de l'utilisateur et retourner les nœuds de document jugés pertinents pour y répondre.

Dans ce chapitre nous commençons par décrire l'annotation au niveau nœud. Nous décrivons ensuite notre approche de reformulation. La dernière section est dédiée à la synthèse et à la conclusion du chapitre.

4.2 Annotation sémantique de nœuds de document

Dans cette section nous présentons tout d'abord le modèle d'annotation *SHIRI-Annot* et la façon avec laquelle les annotations au niveau termes sont transformées au niveau nœud.

1. Une description plus détaillée de cette approche est présentée en section 4.2

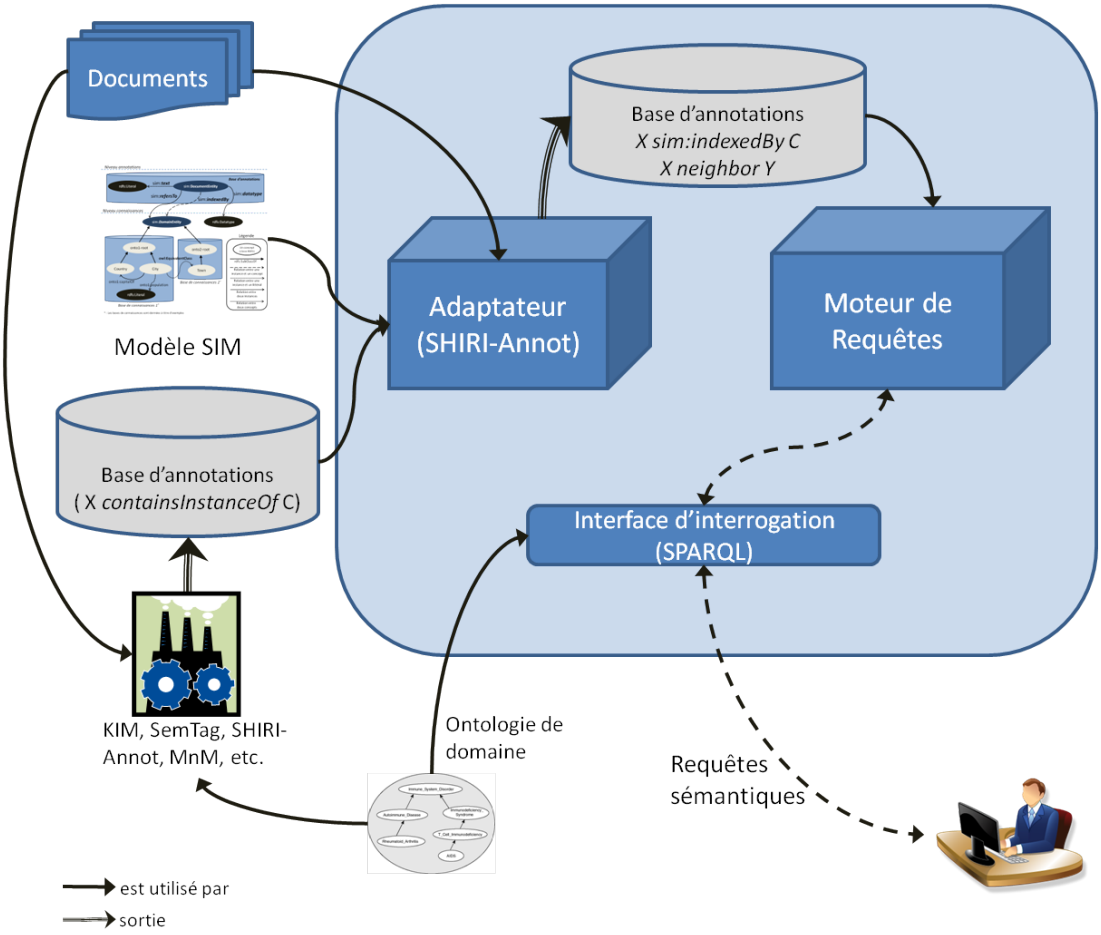

FIGURE 4.1 – Architecture de SHIRI Querying

4.2.1 Description du modèle d'annotation

Par rapport au modèle SIM, le modèle d'annotation proposé par [Thiam *et al.*, 2009] cible des nœuds de documents comme cible et non des entités de document au sens large et définit différents types de nœuds : $\{Node, Concept, SetOfConcept, PartOfSpeech\}$ et la relation binaire *neighbor* pour représenter le voisinage entre deux nœuds.

- Le concept *Node* permet de représenter tous les nœuds de document où une instance de concept a été localisée.
- Le concept *Concept* est utilisé pour annoter un nœud de document contenant seulement une instance de concept. Le concept de cette instance est référencé par la propriété *isIndexedBy*.
- Le concept *PartOfSpeech* est utilisé pour annoter un nœud de document contenant plusieurs instances de différents concepts. Ces concepts sont référencés par la propriété *isIndexedBy*.
- Le concept *SetOfConcepts* est utilisé pour annoter les nœuds de document contenant plusieurs instances de **concepts comparables**. Deux concepts sont comparables s'ils sont liés par la relation *rdfs:subClassOf*. Les différents concepts comparables d'un nœud de type *SetOfConcepts* sont également référencés par la propriété *isIndexedBy*.
- La relation *neighbor* est définie pour représenter la proximité des nœuds dans le document. Deux instances de type $Node$, (n_1, n_2), sont reliées par la relation *neighbor* si :

  1. la distance qui sépare les nœuds de document qu'elles représentent dans l'arbre DOM du document est inférieure à un seuil fixé.
  2. elles sont indexées par au moins un couple de concepts (C_1, C_2) : $\{n_1$ sim:indexedBy $C_1 \wedge n_2$ sim:indexedBy $C_2\}$, tel que C_1 et C_2 sont respectivement domaine et co-domaine d'une propriété R de l'ontologie de domaine.

  La relation *neighbor* est symétrique et non réflexive. Elle est définie comme non reflexive afin d'éviter la redondance de réponses en retrouvant par *neighbor* un nœud qui a déjà été retrouvé grâce aux propriétés *sim:indexedBy* : i.e. il n'est pas utile d'utiliser la relation *neighbor* pour retrouver par quels concepts un nœud donné est indexé.

Le modèle SHIRI-Annot représente aussi les lien d'indexation entre les nœuds de document et les concepts du domaine et les valeurs textuelles des nœuds, que nous ne reprenons pas ici car elles sont déjà exprimées dans le modèle SIM avec d'autres noms (cf. figure 4.2).

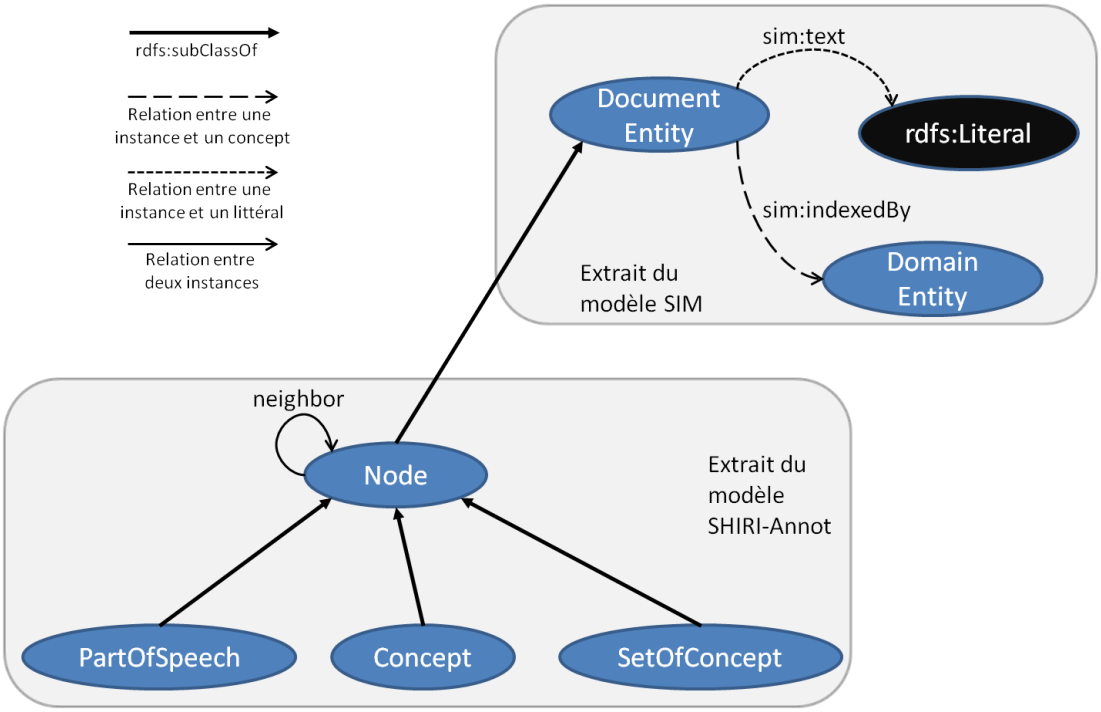

FIGURE 4.2 – Modèle SIM étendu par le modèle SHIRI-Annot

Notation	Signifié
$singleTerm(n)$	Un seul terme a été annoté dans le contenu textuel du nœud.
$containInstanceOf(n, C_o)$	Le nœud contien une instance du concept C_o.
$distance(n_1, n_2)$	La distance entre nœuds correspondant aux instances n_1 et n_2 dans l'arbre DOM du document.
$comparable(n)$	Tous les concepts du domaine qui indexent le nœud n sont comparables.
μ	Seuil de distance de voisinage pour considérer deux nœuds comme étant voisins dans le document

Condition	Inférence
$singleTerm(n) \wedge containInstanceOf(n, C_O)$	$\rightarrow type(n, Concept) \wedge indexedBy(n, C_O)$
$\neg singleTerm(n) \wedge containInstanceOf(n, C_O) \wedge comparable(n)$	$\rightarrow type(n, SetOfConcept) \wedge indexedBy(n, C_o)$
$containInstanceOf(n, C_O) \wedge \neg singleTerm(n) \wedge \neg comparable(n)$	$\rightarrow type(n, PartOfSpeech) \wedge indexedBy(n, C_O)$
$indexedBy(n, C_O) \wedge indexedBy(n', C'_O) \wedge \exists R$ tq. $domain(R, C_o) \wedge range(R, C'_O) \wedge distance(n, n') < \mu$	$\rightarrow neighbor(n, n')$

FIGURE 4.3 – Notations et règles d'annotation $SHIRI - Annot$

Les annotations des nœuds par les classes $Concept$, $SetOfConcept$, $PartOfSpeech$ et les propriétés $neighbor$ et sim:indexedBy se fait par les règles d'annotations

SHIRI-Annot. Ces règles, présentées en figure 4.3 [1], exploitent les annotations des textes des nœuds par concepts du domaine fournies par le système SHIRI-Extract [Thiam *et al.*, 2009] et représentées par la propriété $containInstanceOf$. Il est à noter que la première étape avant l'application de ces règles est de créer des instances de type $Node$ pour représenter les nœuds de document qui ont un contenu textuel annoté par des concepts.

4.2.2 Scénario d'utilisation

La figure 4.4 illustre un exemple d'interrogation de 3 documents annotés par SHIRI-Annot et décrivant des références bibliographiques. Les premières annotations de domaine (e.g. instances des concepts *Person, Topic, Event, Location* et *Affiliation*) sont fournies par différents annotateurs externes. L'adaptateur se charge ensuite d'exploiter les règles d'annotation SHIRI-Annot (cf. figure 4.3) afin d'annoter les nœuds en conformité avec le modèle d'annotation.

Dans ce scénario, le moteur de requêtes reformule une requête utilisateur recherchant les évènements ayant la thématique "Semantic Web". Cette reformulation permet d'atteindre par exemple :

1. Un nœud de type $PartOfSpeech$, indexé par les concepts du domaine *Event* et *Topic* et contenant le terme "Semantic Web" dans son texte. Dans la figure 4.4, il s'agit de la première reformulation SPARQL qui permet d'atteindre le nœud $< p >$ du document 1)

2. deux nœuds voisins de type $Concept$ indexés respectivement par les concepts du domaine *Event* et *Topic*, tel que, le nœud indexé par *Topic* contient la chaîne de caractères "Semantic Web" dans son texte. Dans la figure 4.4, il s'agit de la deuxième reformulation SPARQL qui permet d'atteindre le nœud $< h1 >$ et le deuxième nœud $< li >$ du document 2.

Exemple 1. La requête sémantique q_0 posée par un utilisateur en figure 4.4 est définie par (P_0,F_0,S_0,D) tel que :

1. Les prédicats non décrits ici ont été introduits dans le chapitre 3

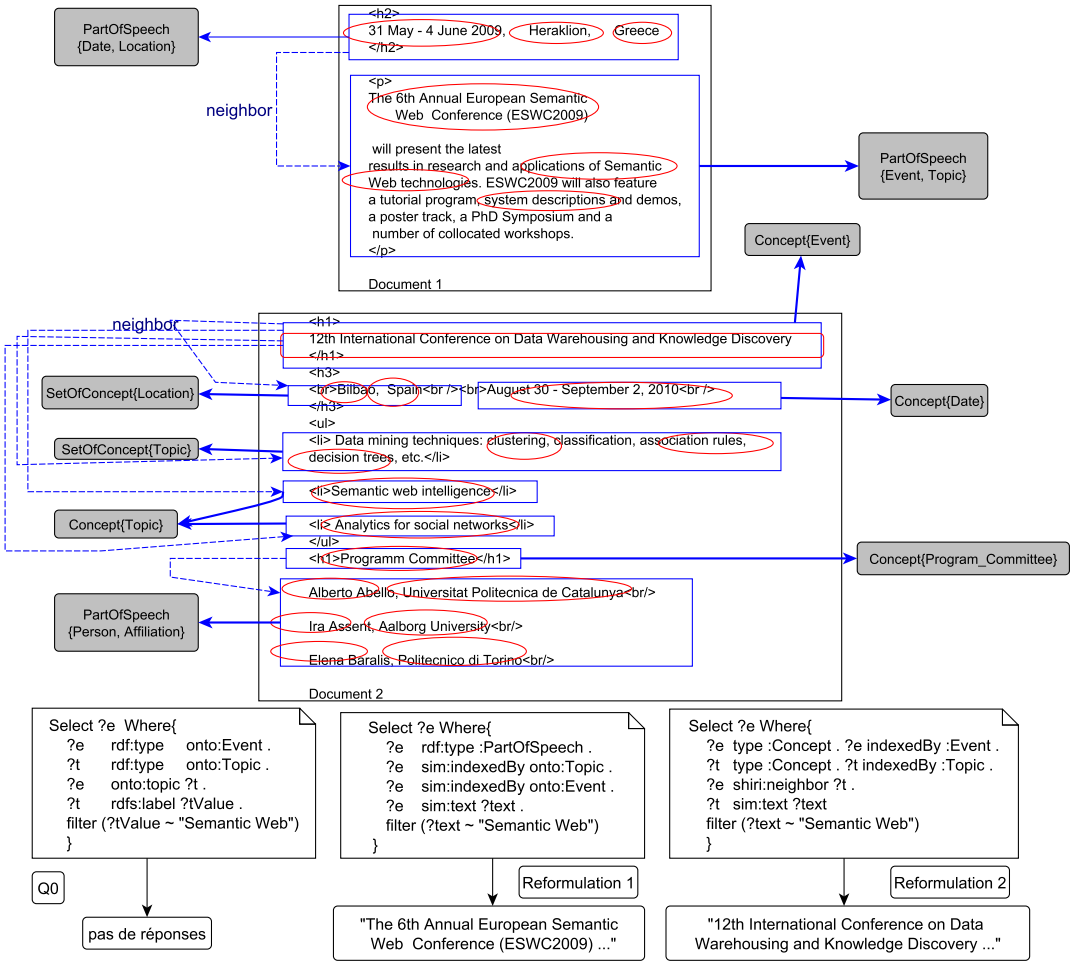

FIGURE 4.4 – Un scénario d'utilisation d'utilisation

P_0 : {
?e rdf:type $Event$.
?t rdf:type $Topic$) .
?e onto:topic ?t .
?t rdfs:label ?$topicValue$.
}
F_0 : $contains$(?$topicValue$,"Semantic Web")
S_0 : { ?e }
D : est l'ensemble des triplets annotant les documents 1 et 2.

Dans la figure 4.4, la requête q_0 n'a pas de réponse : les plateformes d'annotation n'ont pas retrouvé d'instances de concepts ou de relations Le but de la reformulation de q_0 est d'exploiter les annotations générées par l'adaptateur *SHIRI-Annot* pour répondre autrement à la requête utilisateur.

4.3 Reformulation des requêtes

Dans cette section, nous commençons par présenter la première réécriture appliquée à la requête utilisateur pour transformer le problème d'une recherche d'instances ou de faits du domaine en une recherche de nœuds du document. Nous décrivons ensuite les principales heuristiques qui sont employées pour construire des reformulations de cette première réécriture permettant d'atteindre des nœuds de document de différents types. Dans un second temps, nous présentons les reformulations élémentaires proposées pour construire les différentes requêtes reformulées et la relation d'ordre que nous avons définie pour trier ces différentes requêtes. Enfin, nous décrivons l'algorithme $DREQ$ (Dynamic Reformulation and Execution of Queries) qui construit et exécute dynamiquement les reformulations de la requête utilisateur suivant la relation d'ordre définie.

4.3.1 Première réécriture et transformation du problème

Le but de la reformulation est de rechercher des nœuds de document contenant les informations recherchées par l'utilisateur et exprimées à travers sa requête sémantique $Q_0(P_0, F_0, S_0, D)$. Pour cela une première réécriture, notée $neighbor(Q_0)$, est effectuée pour transformer le problème en une recherche de nœuds de documents selon une nouvelle requête $Q_1(P_1, F_1, S_1, D)$.

Dans cette réécriture, tous les typages d'instances sont transformés en une indexation des nœuds recherchés et toute relation entre instances de concepts est

transformée par la recherche d'une relation *neighbor* entre les nœuds indexés par les concepts de ces instances. Les conditions sur les attributs littéraux des instances sont transformées en des conditions sur les contenus textuels des nœuds. Les variables correspondant à des instances ou littéraux retournés par la requête initiale de l'utilisateur, c.à.d., les variables dans S_0, sont remplacées par les nœuds contenant l'instance ou leur contenu textuel. La figure 4.5 montre un exemple pour une telle réécriture à partir d'une requête utilisateur recherchant les thèmes de la conférence "WWW 2011".

	Requête initiale	Requête réecrite
1)	SELECT ?topic WHERE {	SELECT ?node2 WHERE {
2)	?event rdf:type :Event	?node1 sim:indexedBy :Event
3)	?topic rdf:type :Topic	?node2 sim:indexedBy :Topic
4)	?event :hasTopic ?topic	?node1 shiri:neighbor ?node2
5)	?event :hasName "WWW 2011"	?node1 sim:text ?text1
6)	}	FILTER(contains(?text,"WWW 2011"))
7)		}

FIGURE 4.5 – Exemple de première réécriture - Transformation du problème en recherche de nœuds

Cette première réécriture ne sera pas exécutée car elle n'explicite pas le type des nœuds recherchés ($PartOfSpeech$, $SetOfConcept$ ou $Concept$). Elle servira de modèle pour construire plusieurs autres reformulations qui seront triées suivant les types des nœuds recherchés et le nombre de nœuds utilisés pour retrouver la réponse.

Dans notre approche nous partons de l'hypothèse que la requête de l'utilisateur Q_0 est connexe : i.e. il y a au moins un chemin entre deux nœuds du graphe de la requête. Nous considérons les requêtes avec des graphes non connexes comme une union de n requêtes ayant chacune un graphe connexe.

4.3.2 Heuristiques de reformulation

Comme peut le laisser supposer la première réécriture Q_1, plusieurs configurations sont possibles pour retrouver les nœuds indexés par les bons concepts et contenant les valeurs d'attributs recherchées. Les questions qui se posent sont, entre autres : quels types de nœuds recherche-t-on ? Comment combiner des nœuds de types différents dans une même requête ? Combien de nœuds doivent participer à la réponse : va t-on rechercher plusieurs nœuds voisins ou un seul nœud qui agrège toutes les informations demandées ?

Pour répondre à ces différentes questions, nous définissons deux principales heuristiques pour trier les nœuds de documents à retourner suivant leur pertinence par rapport à la requête de l'utilisateur.

La première heuristique consiste à dire que les nœuds moins hétérogènes sémantiquement sont plus précis. Ainsi, les nœuds de type $Concept$, indexés par un seul concept du domaine, seront privilégiés aux nœuds de type $SetOfConcept$, indexés par un ensemble de concepts de domaine comparables, et les nœuds de type $SetOfconcept$ seront privilégiés aux nœuds de type $PartOfSpeech$, indexés par un ensemble de concepts de domaine non comparables.

La deuxième heuristique consiste à privilégier d'abord les réponses constituées d'un seul nœud, puis celles constituées de deux nœuds, puis trois, etc. L'intuition est que plus les éléments sont regroupés, plus ils sont susceptibles d'être liés sémantiquement, sachant que les balises/nœuds de mise en forme sont supprimés des documents XHTML cibles dans un prétraitement que nous effectuons avant l'annotation.

4.3.3 Reformulations élémentaires

Trois reformulations élémentaires sont définies pour construire les requêtes reformulées à partir de Q_1. La reformulation en $Concept$, la reformulation en $SetOfConcept$ et la reformulation en $PartOfSpeech$. Ces reformulations s'appliquent chacune à une partie spécifique de la requête Q_1. Nous définissons ces parties comme étant des patrons de sous-graphes connexes du patron de graphe de la requête initiale, ayant des caractéristiques différentes.

Patron de sous-graphe connexe. Un patron de sous-graphe connexe de P_1, noté g_c, est un sous-graphe de P_1, constitué d'un ensemble de variables $V(g_c) \subset V(P_1)$ t.q. :
– $\forall v_i, v_j$ non littéraux $\in V(g_c)$ il existe au moins un chemin composé d'arcs *neighbor* entre v_i et v_j.
– $\forall v_i \in V(g_c)$, si $< v_i$ sim:text $v_{li} > \in P_1$ alors $< v_i$ sim:text $v_{li} > \in g_c$.

Singleton. Un singleton, noté g_s, est un patron de sous-graphe connexe de P_1, constitué exactement d'une seule variable de type $Node$ et, par conséquent, d'au plus une variable correspondant à un littéral (le texte du nœud unique du graphe).

Patron de sous-graphe connexe comparable. Un patron de sous-graphe connexe comparable de P_1, noté g_{cp}, est un patron de sous-graphe connexe de

P_1 tel que les concepts de g_{cp}, représentés par l'ensemble $C(g_cp)$, sont comparables.

Patron de sous-graphe connexe non comparable. Un patron de sous-graphe connexe non comparable de P_1, noté g_{cn}, est un patron de sous-graphe connexe de P_1 qui n'est pas comparable.

Ainsi, par définition, $g_c = \{g_{cp} \cup g_{cn}\}$ [1]. Sur chaque type de sous-graphe de requête une reformulation élémentaire différente est appliquée.

Reformulation en *Concept*. La reformulation en $Concept$, notée $f_c(g_s)$, s'applique à un singleton g_s de la requête Q_1. Plus précisément, si nous notons $node_i$ la variable de type $Node$ de g_s, cette reformulation explicite le type du nœud recherché en construisant une nouvelle requête par l'ajout du patron de triplet

$node_i$ *rdf:type* shiri:Concept

Le fait de pouvoir agréger plusieurs variables $Node$ de la requête P_1 est essentiel afin d'atteindre les réponses où les informations recherchées sont agrégées dans un même nœud de document. Les reformulations élémentaires en $SetOfConcept$ et $PartOfSpeech$ sont définies dans ce cadre.

Reformulation en *SetOfConcept*. La reformulation en $SetOfConcept$, notée $f_{set}(g_{cp})$, s'applique à un patron de sous-graphe connexe comparable g_{cp} de P_1. Elle consiste à construire une nouvelle requête $Q_s(P_s, S_s, F_s, D)$ à partir de Q_1 en transformant le sous-graphe g_{cp} de P_1 en un singleton g_s avec une variable unique de type $Node$, $node_s$, dans P_s. g_s est construit comme suit :

- $\forall v_i \in g_{cp}$ de type $Node$, si $< v_i$ sim:indexedBy $C_i >\in g_{cp}$ alors $< node_s$ sim:indexedBy $C_i >\in g_s$
- $\forall v_i \in g_{cp}$ de type $Node$, si $< v_i$ sim:text $text_i >\in g_{cp} \wedge \exists e_i(text_i) \in F_1$ alors $< node_s$ sim:text $text_s >\in g_s \wedge e_i(text_s) \in F_1$

Reformulation en *PartOfSpeech*. La reformulation en $PartOfSpeech$, notée $f_{pos}(g_{cn})$, s'applique à un patron de sous-graphe connexe non comparable g_{cn} de P_1. Elle consiste à construire une nouvelle requête $Q_s(P_s, S_s, F_s, D)$ à partir de Q_1 en transformant le sous-graphe g_{cn} de P_1 en un singleton g_s avec une variable unique de type $Node$, $node_s$, dans P_s. g_s est construit comme suit :

- $\forall v_i \in g_{cp}$ de type $Node$, si $< v_i$ sim:indexedBy $C_i >\in g_{cp}$ alors $< node_s$ sim:indexedBy $C_i >\in g_s$
- $\forall v_i \in g_{cp}$ de type $Node$, si $< v_i$ sim:text $text_i >\in g_{cp} \wedge \exists e_i(text_i) \in F_1$ alors $< node_s$ sim:text $text_s >\in g_s \wedge e_i(text_s) \in F_1$

La figure 4.6 montre des exemples pour la première écriture *neighbor* et les différentes reformulations élémentaires appliquées à une requête utilisateur re-

1. Preuve indiquée en annexes

cherchant les événements scientifiques qui ont “semantic web” comme thématique.

La question qui se pose maintenant est comment combiner ces différentes reformulations élémentaires afin (i) d'atteindre toutes les configurations de nœuds de document pouvant répondre à la requête et (ii) d'obtenir les meilleures réponses en premier. Dans la prochaine section, nous présentons notre plan de construction des reformulations qui suit une relation d'ordre que nous avons définie pour mettre en œuvre les heuristiques présentées en section 4.3.2.

4.3.4 Plan de construction des reformulations

La reformulation d'une requête $Q_1(P_1, F_1, S_1, D)$ est une requête $Q_i(P_i, F_i, S_i, D)$ obtenue par la composition des reformulations élémentaires f_c, f_{set} et f_{pos}. Une reformulation Q_i est considérée meilleure qu'une reformulation Q_j si :
- Q_i a moins de variables $Node$ que Q_j
- Q_i a plus de variables de type $Concept$ et le même nombre de variables $Node$ que Q_j.
- Q_i a plus de variables de type $SetOfConcept$, le même nombre de variables $Concept$ et le même nombre de variables de $Node$ que Q_j .

Ce raisonnement est généralisé sur toutes les requêtes reformulées suivant la **relation d'ordre** $\preccurlyeq$.

Une requête Q_i est jugée moins précise que Q_j, $Q_i \preccurlyeq q_j$ ssi :

$(|V(Q_i)| > |V(Q_j)|) \vee ((((|V(Q_i)| = |V(Q_j)|) \wedge$
$(Pos(Q_i) > Pos(Q_j))) \vee ((Pos(Q_i) = Pos(Q_j)) \wedge (Sets(Q_i) \geq Sets(Q_j))))$

avec $Pos(Q_k)$ le nombre de variables de type $PartOfSpeech$ et $Sets(Q_k)$ le nombre de variables de type $SetOfConcept$.

Deux requêtes Q_i et Q_j sont dites de même ordre ssi : $|V(Q_i)| = |V(Q_j)| \wedge Pos(Q_i) = Pos(Q_j) \wedge Sets(Q_i) = Sets(Q_j)$.

Dans ce suit nous présentons l'algorithme de construction et d'exécution de reformulation $DREQ$ qui met en œuvre la relation d'ordre définie pour construire des reformulations de Q_1.

Dynamic Reformulation and Execution of Queries algorithm ($DREQ$)

Les requêtes reformulées sont construites à partir de $Q_1 = neighbor(Q_0)$ en combinant les reformulations élémentaires f_c, f_{set} et f_{pos} sur les différents patrons de sous-graphes connexes de P_1.

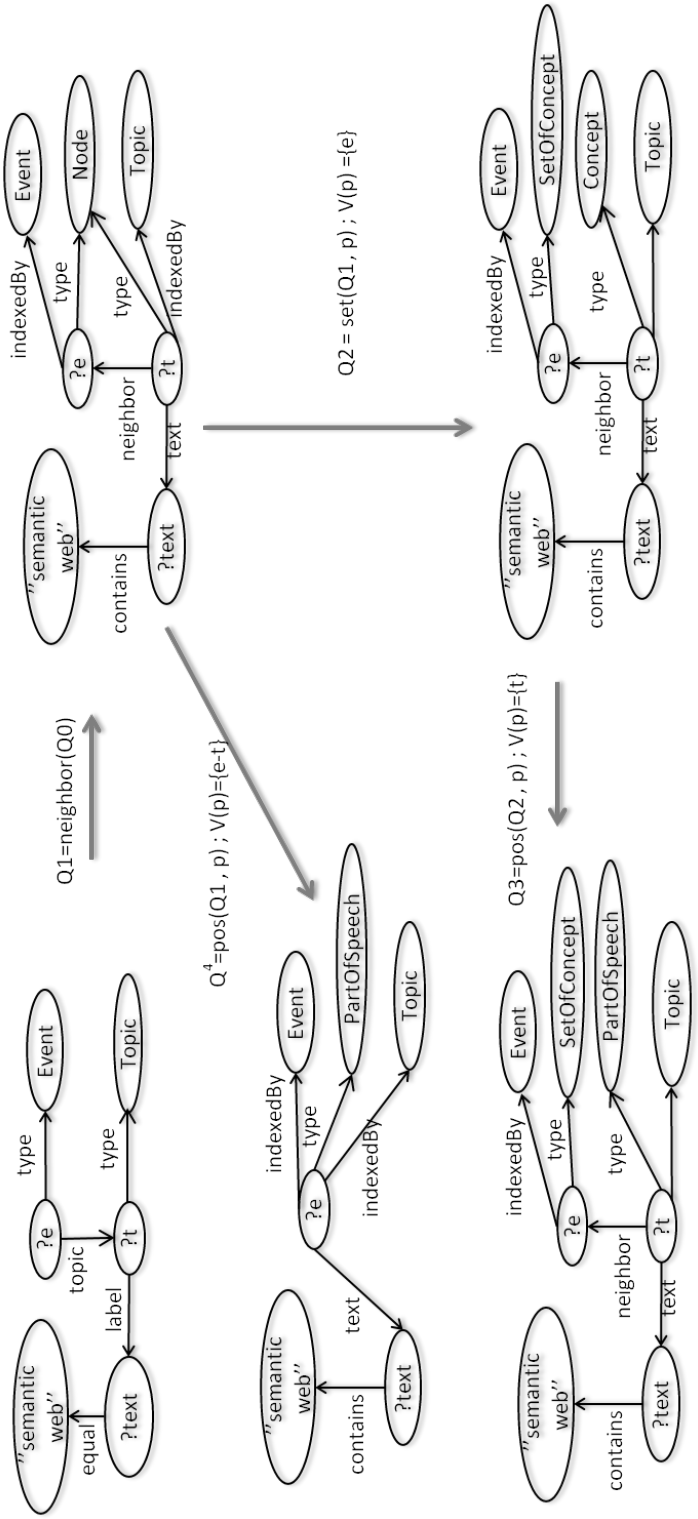

FIGURE 4.6 – Exemples de reformulations élémentaires

Afin de construire toutes les requêtes reformulées possibles suivant la relation d'ordre définie, nous définissons les partitions P_i de P_1 avec i variant de 1 à n ($n = |V(P_1)|$) comme suit :

Une partition P_i de P_1 est un ensemble de patrons de sous-graphes connexes g_{c_1}, g_{c_2}, ..., g_{c_i} reliés par des relations *neighbor* tel que :
– $\forall g_{c_i}, g_{c_j}; V(g_{c_i}) \cap V(g_{c_j}) = \phi$
– $\cup_1^i V(g_{c_i}) = V(P_1)$

Notons que plusieurs partitions P_{ij} différentes sont possibles en fixant le nombre de patrons de sous-graphes connexes à i, sauf pour $i = 1$ et $i = n$ où un seul partitionnement est possible (P_1 et P_n).

Comme la relation d'ordre que nous avons définie privilégie les requêtes avec moins de variables de type $Node$, nous construisons progressivement les partitions P_1 (un seul patron de sous-graphe connexe) puis les partitions P_2 (2 sous-graphes), P_3 jusqu'à P_n en appliquant à chaque partition (i) les reformulations f_c et (ii) les reformulations f_{set} et f_{pos} sur les patrons de sous-graphes connexes (non) comparables qui ne sont pas des singletons. L'application des reformulations f_{set} et f_{pos} sur les singletons est effectuée ultérieurement à un autre niveau d'ordre tel que fixé par la relation d'ordre définie.

Aussi, comme les variables de type $Concept$ sont préférées à celle du type $SetOfConcept$ et $PartOfSpeech$, les partitions P_{ij} qui ont un même nombre de sous-graphes connexes i sont triées suivant le nombre de sous-graphes singletons, l_{ij}, générés par la partition.

Les partitions sont construites d'une manière incrémentale en effectuant une recherche en largeur d'abord, telle que présentée à la figure 4.7, en supprimant virtuellement les liens *neighbor* un à un. À chaque fois qu'une relation *neighbor* est exploitée pour créer des partitions, les patrons de sous-graphe connexes sont calculés et ajoutés pour chaque partition P_{ij} suivant les valeurs de i et de l_{ij}, sachant que chaque partition est unique.

Une fois que les partitions sont construites et triées, les reformulations élémentaires sont appliquées aux différents patrons de sous-graphes connexes suivant le type de ces derniers (cf. algorithme 1). Nous notons respectivement f_c^k, f_{set}^k, f_{pos}^k les reformulations élémentaires en $Concept$, $SetOfConcept$ et $PartOfSpeech$ appliquées à k patrons de sous-graphes connexes respectivement des singletons, des comparables et des non comparables.

Ainsi, le nombre de reformulations pour une partition P_{ij} est $3^{l_{ij}}$. Le nombre de toutes les partitions possibles de P_1 est maximal si P_0 est un patron de graphe RDF élémentaire complet. Cela correspond au nombre de Bell $B_n = \sum_{k=0}^{n-1} C_{n-1}^k B_k$

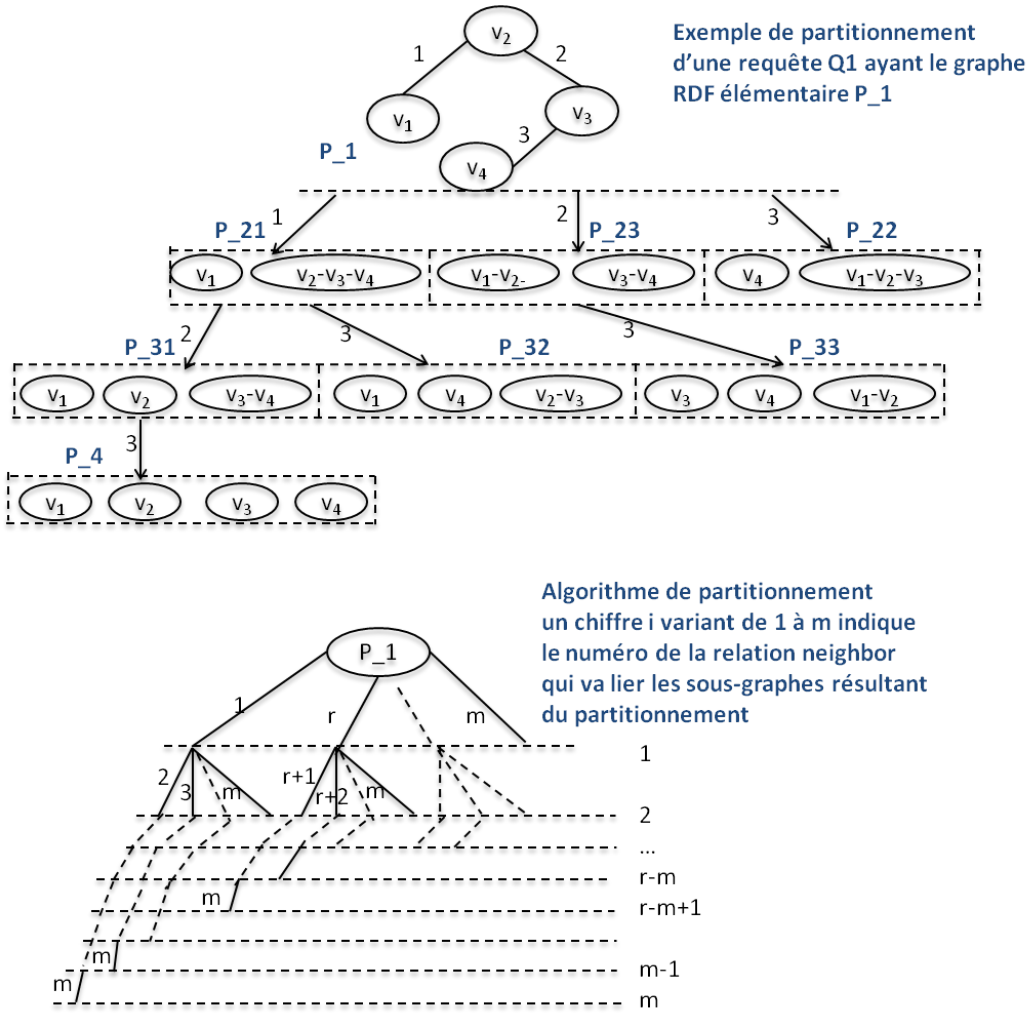

FIGURE 4.7 – Algorithme de partitionnement

avec $B_0 = B_1 = 1$. Par exemple, dans la figure 4.7, le nombre de partitions est de 7 (par comparaison au pire cas avec $B_4 = 15$ partitions) et le nombre de reformulations est de 92.

Il est clair que la complexité du partitionnement et de la reformulation est exponentielle par rapport au nombre de variables dans la requête de l'utilisateur [1], cependant, dans la pratique, P_1 est rarement un graphe complet et le nombre de variables de la requête utilisateur ne dépasse pas les 10 variables dans la majeure partie des cas. Aussi, un avantage important est que l'algorithme *DREQ* peut être arrêté à un seuil fixé grâce à la relation d'ordre moyennant les valeurs

1. Calculs de complexité présentés en annexes

Algorithme 1 Algorithme de reformulation DREQ

```
1  Début
2      Pour i ∈ 1...n Faire
3          Pour P_ij : j ∈ 1...j_max Faire
4              l^s_ij = {g_cp de P_ij}
5              l^p_ij = {g_cn de P_ij}
6              générer et exécuter les requêtes reformulées f^{l^s_ij}_set ∘ f^{l^p_ij}_pos ∘ f^{l_ij}_c
7              Pour l ∈ (l_ij − 1)...0 Faire
8                  générer et exécuter les requêtes reformulées f^{l^s_ij + l_ij − l}_set ∘ f^{l^p_ij}_pos ∘ f^{l}_c
9                  générer et exécuter les requêtes reformulées f^{l^s_ij}_set ∘ f^{l^p_ij + l_ij − l}_pos ∘ f^{l}_c
10 Fin
```

i et l (cf. algorithme 1) et permettre de retourner les réponses retrouvées par les meilleures reformulations.

4.4 Conclusion

Dans ce chapitre, nous avons présenté notre approche, SHIRI-Querying, pour l'interrogation sémantique de documents semi-structurés annotés par des instances de concepts. Cette approche permet de construire des reformulations de la requête utilisateur permettant de retourner des nœuds de document en réponse aux requêtes sémantiques posées. SHIRI-Querying utilise les règles d'annotation de SHIRI-Annot ([Thiam *et al.*, 2009]) pour annoter les nœuds de document à partir des annotations de leur contenu textuel par des concepts. Cette transformation offre la possibilité de retourner des réponses constituées de nœuds de document et de raisonner sur la composition sémantique des nœuds, là ou la requête initiale de l'utilisateur n'aurait obtenu aucune réponse.

L'approche SHIRI-Querying a plusieurs applications possibles. Elle peut aussi bien s'inscrire dans la problématique de recherche de passages que dans celle des systèmes de questions-réponses. Des experts de divers domaines d'application pourront notamment utiliser l'approche pour construire semi-automatiquement une base de connaissances à partir des nœuds de document retournés par *SHIRI-Querying*. L'évaluation de l'approche, que nous proposons au chapitre suivant, s'inscrit dans la problématique des systèmes de questions-réponses où des utilisateurs ou des interfaces construisent et soumettent des requêtes sémantiques

fondées sur une ontologie de domaine.

Cette évaluation est effectuée sur deux corpus extraits du Web. Deux points sont particulièrement évalués : (i) la précision des nœuds retournés et (ii) l'efficacité du tri des réponses par la relation d'ordre proposée dans ce chapitre.

Chapitre 5

Évaluation de l'approche *SHIRI-Querying*

Dans ce chapitre nous évaluons l'approche *SHIRI-Querying* sur deux principaux plans : (i) la précision des nœuds retournés et (ii) l'efficacité du tri des réponses par le tri des reformulations proposées dans le chapitre 4. Nous commençons par préciser les critères d'évaluation pris en compte et les types d'erreur que nous pouvons rencontrer. Nous présentons ensuite le premier corpus d'expérimentation et les résultats obtenus, puis le deuxième corpus d'expérimentation et les résultats obtenus dans son cadre. Enfin nous proposons une discussion et une synthèse de l'approche à la lumière de ces résultats avant de conclure.

5.1 Critères et mesures d'évaluation

L'évaluation des systèmes de recherche sémantique n'est pas une tâche évidente. Nous précisons dans ce qui suit les entrées et sorties du système ainsi que les critères de jugement des réponses.

Les entrées du système sont des requêtes sémantiques, telles que définies dans le chapitre 3. Les données sont des annotations RDF de documents XML et/ou

HTML. Les réponses proposées aux requêtes sont des IRIs d'instances du concept *Node* correspondant à des nœuds de document.

Une réponse est jugée correcte si les nœuds retournés contiennent effectivement des références aux instances de concepts du domaine recherchées par l'utilisateur et aux relations sémantiques exprimées dans la requête. Par exemple, si l'utilisateur recherche la conférence $CAiSE'2010$ et sa date, les nœuds contenant cette conférence et la date d'un de ses workshops seront considérés comme de fausses réponses.

La figure 5.1 montre le logiciel que nous avons développé pour poser les requêtes sémantiques et évaluer les réponses de chaque reformulation. Une fenêtre de navigation nous permet de cliquer sur chaque réponse afin de surligner les nœuds de documents qui la constituent.

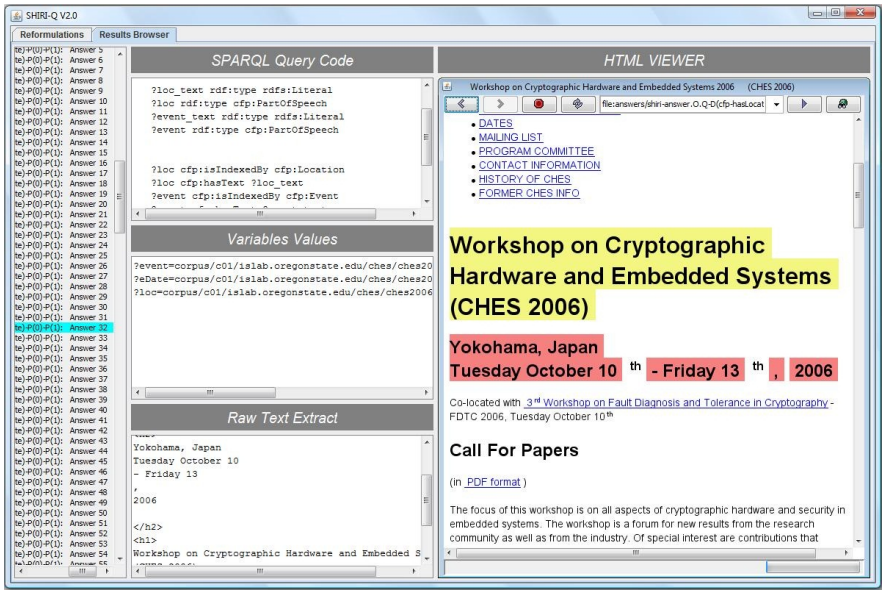

FIGURE 5.1 – Interface de validation des réponses du logiciel *SHIRI-Querying*

Dans le cadre de ces expérimentations, la relation *neighbor* lie deux instances du concept *Node* si les nœuds de document considérés sont séparés par un chemin non orienté de longueur $\leq d$ dans l'arbre XML ou HTML. Aussi, comme nous souhaitons uniquement évaluer les performances de l'approche de reformulation, nous prenons seulement en compte les annotations sémantiques correctes, i.e. les réponses comportant des bruits – générés par les systèmes d'annotation tiers – ne sont pas prises en compte.

5.2 Premier corpus

Nous avons évalué notre approche sur deux corpus différents [1]. Le premier corpus regroupe les annotations RDF d'extraits de trois sources de données bibliographiques (DBLP, HAL et serveur interne de l'INRIA). Les annotations contiennent au départ des instances des concepts *Article, Conférence, Date, Lieu, Personne* et les relations/attributs relatifs (ex. *écritPar, publiéDans, lieu, date, titre*). L'ensemble des annotations constitue près de 10.000 triplets RDF.

Nous avons soumis un ensemble de 5 requêtes pour rechercher des conférences et éventuellement leurs dates, lieux, articles et auteurs correspondants. Le petit nombre de requêtes se justifie par rapport à la régularité de structuration des documents concernés et au nombre des concepts/relations avec lesquels ils ont été annotés. Un des buts était aussi d'étudier la capacité de l'approche à intégrer différentes sources de données au moment de l'interrogation.

La figure 5.2 présente le rappel et la précision des réponses quand le seuil de la distance de voisinage d varie de 1 à 7. Les résultats montrent que pour $d \leq 3$, toutes les réponses sont atteintes avec 100% de précision. Mais si le seuil de distance est ≥ 4, la précision est de presque 0%. En effet, dans deux sources de données, chaque article est associé à toutes les conférences pour ce seuil.

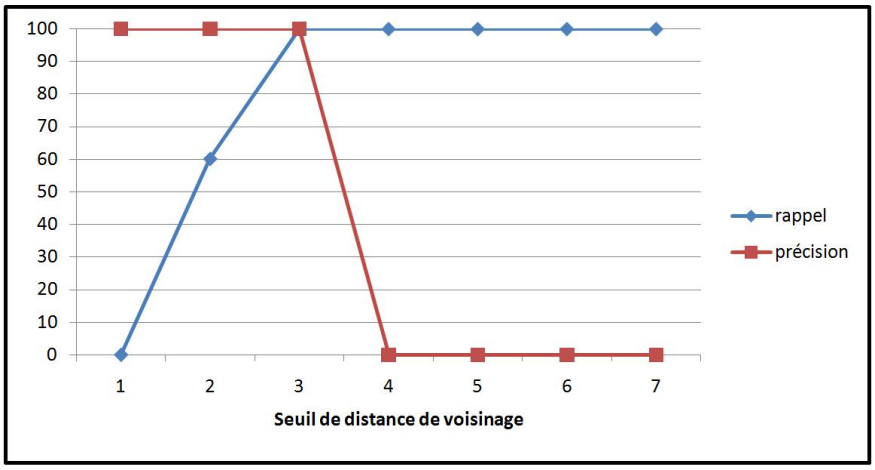

FIGURE 5.2 – Rappel et précision en fonction de la distance de voisinage d

La figure 5.2 présente le rappel et la précision obtenus pour $d \leq 4$ et un seuil d'ordre variant de 1 à 10 pour les requêtes reformulées, 10 étant l'ordre de reformulation maximum atteint pour les requêtes utilisateur soumises. À un seuil

1. http ://wwwdi.supelec.fr/~ bennacer/SHIRI/datasets.html

d'ordre i donné, les premiers (meilleurs) i ensembles de reformulations sont générés et exécutés par *DREQ*. Les résultats montrent que la précision diminue lorsque le seuil d'ordre de reformulation augmente. Un rappel de 100% est atteint pour toutes les requêtes soumises après le 9^{eme} seuil d'ordre. D'un autre côté, chacune des sources de documents a une structure propre mais nous avons pu récupérer les réponses de toutes les sources par la même interrogation grâce à l'adaptation des annotations au niveau nœuds et à la reformulation des requêtes utilisateurs.

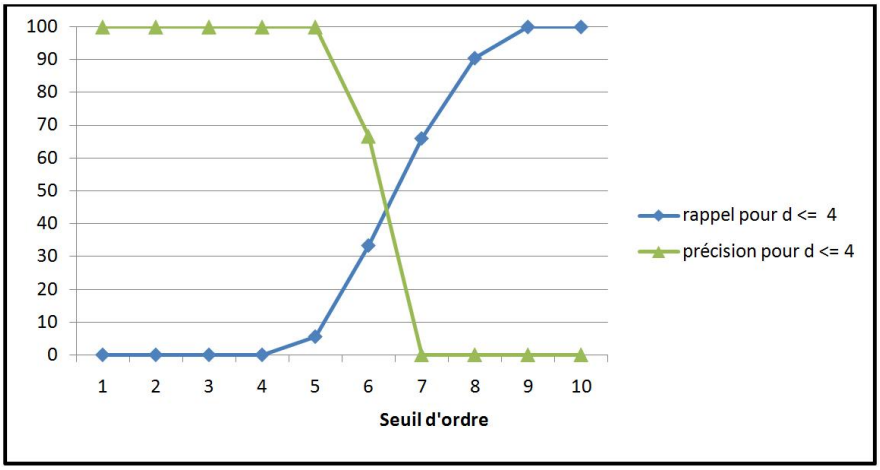

FIGURE 5.3 – Rappel et précision en fonction du seuil d'ordre des reformulations

5.3 Deuxième corpus

Le deuxième corpus est constitué de 32 sites d'appels à communications annotés (environ 30,000 triplets RDF). Les annotations des termes par des concepts du domaine ont été générées par *SHIRI-Extract* [Thiam *et al.*, 2008]. Elles concernent des noms de conférences, des lieux, des dates, des membres de différents comités, ou des thèmes de recherche. Ces annotations ont ensuite été automatiquement transformées par l'adaptateur de *SHIRI-Querying* pour se conformer au modèle d'annotation niveau nœud. Enfin, nous avons soumis 15 requêtes[1] sémantiques suivant l'ontologie *Call-For-Paper* [Thiam *et al.*, 2009].

Aucune des requêtes de domaine soumises n'a de réponses à cause de l'absence d'instances de concepts et de relations. Sans reformulation, le rappel est donc de 0% si on considère les informations présentes dans les documents. Avec notre

1. Les listes de requêtes sont fournies en annexe

approche nous avons pu atteindre un rappel total de 56% pour une distance $d \leq 7$. Étant donné la grande hétérogénéité du corpus, ce rappel a été mesuré au pire cas, e.g. si la requête demandait le lieu d'un événement, nous avons considéré que tous les événements référencés dans la base avaient bien l'information lieu correspondante dans les documents.

La figure 5.3 décrit la précision des réponses quand le seuil de distance de voisinage d varie de 1 à 7. Sur ce corpus, la précision est au dessus de 72% pour une distance $d \leq 7$. En dépit du fait que les valeurs de seuil pertinentes pour d varient d'un corpus à un autre, notre expérimentation a bien validé (i) l'hypothèse selon laquelle des relations sémantiques peuvent être retrouvées entre deux nœuds voisins dans des documents plus ou moins hétérogènes et (ii) l'efficacité des heuristiques de reformulation et de tri adoptées dans *SHIRI-Querying* pour retouner les nœuds de document les plus précis en premier.

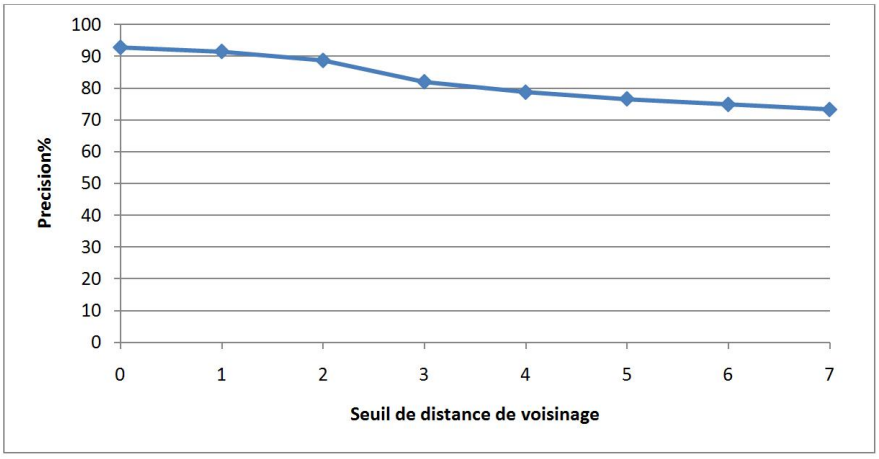

FIGURE 5.4 – Précision en fonction de la distance de voisinage d

5.4 Synthèse et discussion

La figure 5.3 présente la précision moyenne des réponses pour les mêmes requêtes et pour plusieurs valeurs de seuil différentes pour d. La valeur d'ordre varie de 1 à 18, 18 étant l'ordre maximum atteint pour les requêtes soumises. Les résultats montrent que la précision diminue bien au fur et à mesure que le seuil d'ordre de la reformulation augmente.

Par exemple, la date de la conférence $CHES$ est correcte quand elle est localisée dans le nœud $PartOfSpeech$ qui contient la conférence et son nom, mais fausse

FIGURE 5.5 – Précision en fonction du seuil d'ordre des reformulations

quand elle est localisée dans un nœud $PartOfSpeech$ voisin. Les thèmes de la conférence sont aussi corrects s'ils sont localisés dans un nœud $SetOfConcept$ mais faux quand ils sont localisés dans un nœud $PartOfSpeech$ voisin, indexé par plusieurs concepts différents. En contre partie, nos reformulations apportent aussi de fausses réponses puisque elles reposent à la base sur des méthodes heuristiques. Cependant, nous avons fortement limité l'effet de ces erreurs avec la distance de voisinage et le seuil d'ordre des reformulations.

Du point de vue de l'exécution des requêtes, nous employons une optimisation simple qui consiste à parcourir la base d'annotations avant toute interrogation afin d'identifier les types de nœuds inexistants. Cette étape réduit considérablement le nombre de reformulations (e.g. il n'y a pas de nœuds dans le corpus que nous avons utilisé). Sur les requêtes soumises, le temps de reformulation et d'exécution moyen est de 483 ms sans optimisation et 266 ms avec optimisation, sur une machine avec un processeur Core 2 Duo T9300 et 4Gb de RAM.

5.5 Conclusion

Les résultats expérimentaux obtenus sont prometteurs. Ils montrent que le rappel augmente et que la précision diminue raisonnablement au fur et à mesure de la construction et de l'exécution ordonnée des requêtes reformulées.

Cependant le cadre étudié ici est celui ou aucune base de connaissances n'a pu être exploitée et où les instances de concepts sont soit absentes, soit crées à la volée, sans autre description que leur concept. Dans les chapitres suivants,

nous étudions le cas où des bases de connaissances sont exploitables et où les instances de concept du domaine utilisées pour l'annotation ont une description en termes de relations et d'attributs. Nous proposons dans ce cadre une approche pour retrouver des réponses nouvelles aux requêtes sémantiques en enrichissant en amont les bases de connaissances disponibles ou les bases de connaissances issues de l'annotation.

Chapitre 6

Enrichissement contrôlé de bases de connaissances à partir de documents semi-structurés

6.1 Introduction

De plus en plus de bases de connaissances RDF sont disponibles sur le Web (e.g. le projet Linked Open Data regroupant au moins 25 milliards de triplets [1]). Ces bases de connaissances ne sont pas complètes et beaucoup d'informations restent

1. http://www.semantic-web-journal.net/content/speeding-disk-rdf-index-lookups-using-bhash-trees

disponibles dans les documents. Alors que de plus en plus d'outils permettent d'annoter sémantiquement les documents semi-structurés, l'extraction automatisée de relations sémantiques reste un défi de taille comparé à l'extraction des instances de concepts. Les approches d'extraction de relations existantes dans la littérature supposent en général l'existence de patrons lexico-syntaxiques ou de régularités structurelles dans les documents. Ces hypothèses ne sont souvent pas vérifiées pour des documents qui proviennent de sources différentes et qui sont structurellement hétérogènes.

Dans cette partie, nous proposons une approche appelée REISA (contRoled Extension and Interrogation of Semantic Annotation) qui génère des instances de relations sémantiques candidates à partir de documents annotés par des instances de concepts. Nous nous plaçons dans le contexte d'un domaine d'application cible pour lequel nous disposons :

– d'un corpus de documents semi-structurés pertinents par rapport au domaine d'étude,
– d'un ensemble d'outils d'annotation sémantique qui permet de retrouver des références aux instances de concepts dans le corpus cible,
– d'une ou plusieurs bases de connaissances RDF préexistantes décrites conformément à des ontologies de domaine éventuellement communes ou alignées.

Le premier objectif de notre approche est d'enrichir les bases de connaissances interrogées en proposant de nouvelles instances de relations sémantiques candidates entre les instances de concepts annotant des parties de documents voisines. Cet enrichissement se fait en exploitant la structure des documents annotés, les résultats des outils d'annotation par instances de concepts et les bases de connaissances RDF disponibles.

Plus précisément, la structure des documents XHTML peut aider à délimiter un espace de recherche des instances de relations, même en l'absence de régularités connues. En effet, plus les entités de document référant à des instances de concepts sont proches dans un document, plus elles sont susceptibles d'être liées sémantiquement. Cependant, l'utilisation d'une heuristique aussi simple risque de générer de nombreux candidats erronés. Dans notre approche, nous exploitons la sémantique des ontologies et les instances présentes dans les bases de connaissances préexistantes pour contrôler au mieux les instances de relations candidates.

Le deuxième objectif de notre approche est d'interroger conjointement les connaissances produites par enrichissement et les bases de connaissances préexistantes ou issues de l'annotation des documents de manière transparente. Étant donné que ces bases de connaissances sont construites avec des méthodes différentes, il est important de distinguer les faits suivant les techniques qui ont conduit à leur

construction et de représenter la confiance qui leur est attribuée. Cette représentation, prise en compte par le modèle SIM à travers les graphes nommés RDF, permet d'améliorer la recherche d'information en triant les réponses aux requêtes posées par les utilisateurs.

Dans la section 6.2, nous présentons une description générale de l'approche REISA. En section 6.4, nous présentons notre méthode d'enrichissement. Enfin, notre méthode d'interrogation est décrite en section 6.5.

6.2 Description générale de l'approche

L'approche REISA (*cont**R**oled **E**xtension and **I**nterrogation of **S**emantic **A**nnotations*) enrichit les connaissances qui seront interrogées. Ce processus passe par une représentation homogène des connaissances exploitées par la méthode d'enrichissement. L'approche s'intéresse, par ailleurs, au processus d'interrogation de l'ensemble des connaissances disponibles, intégrant celles provenant de l'enrichissement. La figure 6.1 illustre l'approche REISA et ces trois activités principales. Le rôle et les entrées/sorties de chaque activité sont décrits dans cette section.

6.2.1 Intégration

Le rôle principal de l'intégration consiste à mettre les annotations et les faits des bases de connaissances RDF préexistantes en conformité avec le modèle sémantique SIM que nous avons défini (cf. chapitre 3).

L'annotation des documents fournit des liens entre les parties de documents annotées et des instances de concepts ou de relations découvertes par annotation ou représentées dans les bases de connaissances préexistantes.

Plusieurs outils d'annotation sémantique et automatique des documents sont disponibles mais il n'existe pas de convention pour régir le format de sortie des annotations produites. Chaque système propose souvent son propre format de sortie (e.g. chaînes de caractères au format JSON pour KIM, triplets RDF pour SOFIE). L'activité d'intégration a, entre autres, pour rôle de représenter ces différentes sorties au format RDF.

Des instances d'entités de documents sont créées en respectant la granularité de l'annotateur (e.g. terme, nœud) et en établissant les liens *sim:refersTo* entre l'entité de document et les instances de concepts utilisées pour l'annotation. Les instances d'entités de document, leur contenu textuel et leurs liens avec les entités

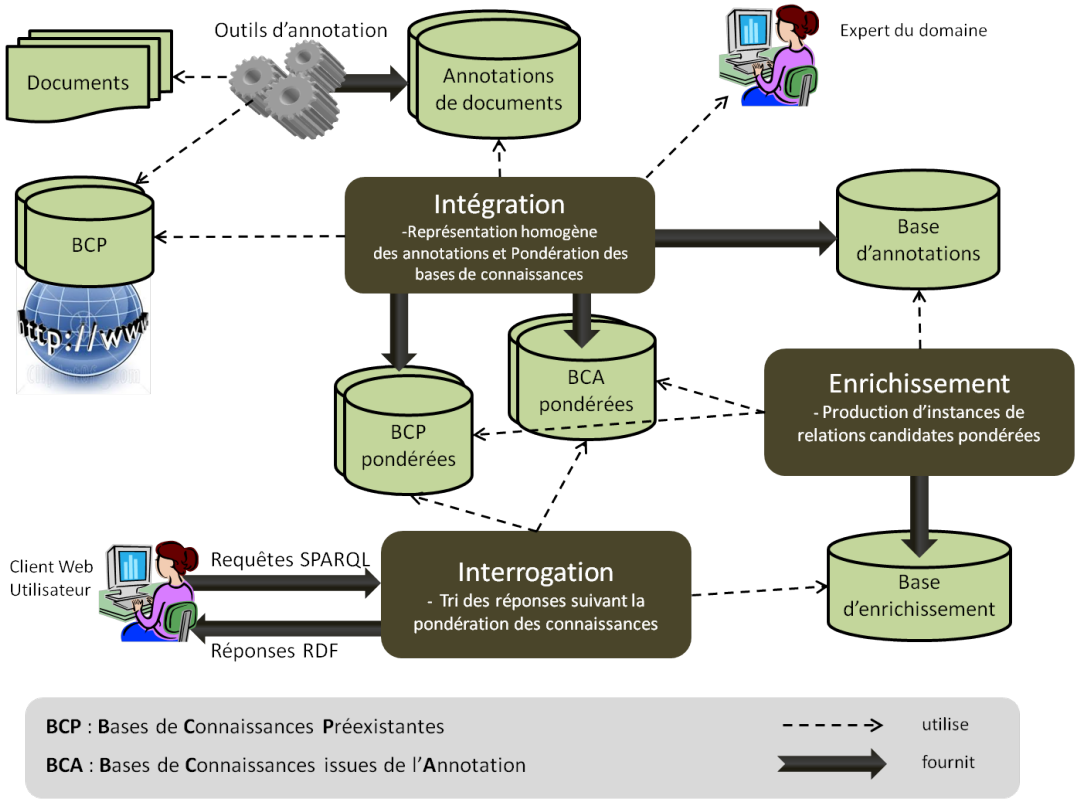

FIGURE 6.1 – Illustration de l'approche REISA

de domaine sont sauvegardées dans la base d'annotations. Les nouvelles instances découvertes par les annotateurs sont regroupées dans une base de connaissances dédiée, notée BCA.

L'intégration consiste aussi à affecter une valeur de confiance aux faits en tenant compte de leur provenance. Cette pondération est effectuée par un expert du domaine. L'exemple décrit dans la figure 6.4 montre le résultat du processus d'intégration sur des extraits de bases de connaissances et d'annotations (cf. figure 6.3) de documents du domaine géographique. La figure 6.2 contient l'extrait de document semi-structuré annoté par KIM. Dans cet exemple, la base de connaissances préexistante (cf. figure 6.4) contient des informations relatives au Laos, sa capitale et son continent. La base d'annotations comprend les annotations d'une partie de document (nœud XML) contenant des références au Laos, au Vietnam, au Mékong et à Hanoi. La relation *sim:gweight* associe une confiance plus élevée à la base de connaissances préexistante (BCP) utilisée par KIM (poids de 1) qu'à BCA (poids de 0.9) dont les faits ont été extraits des documents par des procédés automatiques. Les poids associés aux différents graphes nommés sont donnés ici à titre indicatif.

Notons que l'étape d'intégration produit une base de connaissances préexistante unique dont l'ontologie est l'union de toutes les ontologies de domaine considérées. Cette intégration est effectuée en prenant en compte les éventuels alignements connus (relations OWL) entre les bases de connaissances préexistantes considérées.

6.2.2 Enrichissement

L'activité d'enrichissement produit de nouvelles instances de relations sémantiques et leur associe une mesure de confiance. Elle exploite pour cela quatre éléments :

- les bases de connaissances préexistantes,
- la base de connaissance issue de l'annotation (BCA),
- la base d'annotations,
- la structure des documents annotés.

En sortie, cette activité fournit une base de connaissances, dite base d'enrichissement. Cette base regroupe les instances de relations candidates dans différents graphes nommés suivant une estimation de leur fiabilité (poids) calculée automatiquement. La figure 6.5 illustre un exemple de base d'enrichissement constituée de trois graphes nommés de différents poids.

...

Laos traces its history to the kingdom of Lan Xang (Million Elephants), founded in the 14th century by a Lao warlord who took over **Vientiane** with 10,000 Khmer troops. Within 20 years of its formation the kingdom expanded eastward to Champa and along the Annamite mountains in **Vietnam**.

Stone tools discovered in northern **Laos** attest to the presence of hunter-gatherers from at least 40,000 years ago. From the fourth to the eighth century, communities along the **Mekong** River began to form into townships, or Muang as they were called.

...

Following the military defeat of Japan and the fall of its puppet Empire of Vietnam in August 1945 the Viet Minh occupied **Hanoi** and proclaimed a provisional government which asserted national independence on 2 September.

...

FIGURE 6.2 – Extrait de document annoté

Annotations de documents Résultats pour l'annotation de l'extrait de document avec KIM
{ term="Laos" class="Country" instance="kimkb :Laos.0" }
{ term="Vietnam" class="Country" instance="kimkb :Vietnam.0" }
{ term="Mekong" class="River" instance="kimkb :Mekong.0" }
{ term="Hanoi" class="City" instance="kimkb :Hanoi.0" }

FIGURE 6.3 – Extrait des annotations RDF en entrée

6.2.3 Interrogation

Afin d'exploiter les divers types de connaissances dont nous disposons, les requêtes utilisateur, formulées avec le vocabulaire d'une ontologie d'une des bases de connaissances interrogées, sont réécrites dans l'objectif de trier les réponses suivant leur poids.

Plus précisément, il s'agit de prendre en entrée une requête SPARQL, de la réécrire en explicitant les graphes nommés correspondant à chaque triplet de la requête et en utilisant une fonction pour trier les réponses suivant les poids des graphes. La figure 6.6 présente un exemple de cette reformulation pour une requête en entrée demandant la liste des pays avec leur capitale.

Dans cet exemple, la fonction avg (moyenne) est utilisée pour trier les réponses

```
@prefix graphs: <http://reisa.com/graphs/>
graphs:knowledgebase = {
kimkb:Laos.0 rdf:type onto:Country
kimkb:Laos.0 onto:partOf kimkb:Continent.2
kimkb:Continent.2 rdf:type onto:Continent
kimkb:Continent.2 rdfs:label "Asia"
kimkb:Vientiane.0 rdf:type onto:City
kimkb:Vientiane.0 onto:capital kimkb:Laos.0
}
graphs:annotationsbase = {
  corpus:doc0/html/body/div/p[3]/a.0 rdf:type sim :DocumentEntity
  corpus:doc0/html/body/div/p[3]/a.0 sim:refersTo kimkb:Vietnam.0
  corpus:doc0/html/body/div/p[3]/a.0 sim:text "Vietnam"
  corpus:doc0/html/body/div/p[3].12 sim:refersTo kimkb:Laos.0
  corpus:doc0/html/body/div/p[3].12 sim:text "Laos"
  corpus:doc0/html/body/div/p[3].20 sim:refersTo kimkb:Mekong.0
  corpus:doc0/html/body/div/p[3].20 sim:text "Mékong"
  corpus:doc0/html/body/div/p[4]/a[2].0 sim:refersTo kimkb:Hanoi.0
  corpus:doc0/html/body/div/p[4]/a[2].0 sim:text "Hanoi"
}
graphs:knowledgebase sim:gweight 1
graphs:annotationsbase sim:gweight 0.9
```

FIGURE 6.4 – Exemple de base de connaissances et d'annotations après intégration

```
@prefix graphs: <http://reisa.com/graphs/>
graphs:candidates.dbpedia.country.0 {
kimkb:Mekong.0 dbpedia:country kimkb:Laos.0
}
graphs:candidates.dbpedia.country.1 {
kimkb:Mekong.0 dbpedia:country kimkb:Vietnam.0
}
graphs:candidates.kim.capital.2 {
kimkb:Hanoi.0 kim:capital kimkb:Vietnam.0
}
graphs:candidates.dbpedia.country.0 sim:gweight 0.8
graphs:candidates.dbpedia.country.1 sim:gweight 0.7
graphs:candidates.kim.capital.2 sim:gweight 0.6
```

FIGURE 6.5 – Exemple de triplets d'enrichissement

suivant leur poids mais d'autres fonctions d'agrégation peuvent être employées.

SELECT ?c ?t WHERE {
?c *rdf* : *type kim* : *Country*
?t *rdf* : *type kim* : *City*
?t *kim* : *capital* ?c
}
	SELECT ?c ?t WHERE {
	GRAPH ?g1 { ?c *rdf* : *type kim* : *Country* }
	GRAPH ?g2 { ?c *rdf* : *type kim* : *City* }
	GRAPH ?g3 { ?t *kim* : *capital* ?c }
	?g1 *sim* : *gweight* ?p1
	?g2 *sim* : *gweight* ?p2
	?g3 *sim* : *gweight* ?p3
	} ORDER BY avg(?p1, ?p2, ?p3)

FIGURE 6.6 – Exemple de reformulation d'une requête SPARQL

6.3 Pondération des bases de connaissances

Dans le cadre de notre approche, nous souhaitons distinguer les connaissances selon leur degré de confiance. Cette information sera, pour nous, un critère de tri des réponses aux requêtes posées par les utilisateurs. Ce degré de confiance est affecté globalement à un ensemble de faits suivant deux critères : leur provenance et leur technique de construction. Dans notre approche, la pondération concerne uniquement les ensembles de faits décrivant les instances de domaine. Les faits ayant une même valeur de pondération sont regroupés dans un même graphe nommé RDF ([Carroll *et al.*, 2005]). Les ontologies sont supposées toujours valides.

Un **Graphe RDF Nommé** est un graphe RDF identifié par une URI [1]. Tel que mentionné dans la spécification $SPARQL$, un **ensemble de données RDF** (dataset) est une collection de graphes. Il comprend un graphe (le graphe par défaut) qui n'est pas nommé et optionnellement plusieurs graphes nommés [2].

Dans le modèle SIM, le concept *sim:NamedGraph* est la classe des graphes RDF nommés qui permet de représenter la pondération des faits appartenant aux différentes bases de connaissances. Chaque graphe nommé a un poids dont la valeur est comprise entre 0 et 1. Ce poids est représenté par l'attribut *sim:gweight*. Les URIs des graphes nommés et leur pondération sont sauvegardées dont le graphe RDF par défaut.

Dans le cadre de notre approche, nous distinguons deux techniques de pondération :

Première technique de pondération. Il s'agit d'une pondération manuelle par l'expert du domaine qui associe un poids aux bases de faits ($T^j_{O^i}$) des bases de connaissances BC^j. Nous distinguons ici deux niveaux de confiance :

1. Un premier niveau caractérise les bases de connaissances sûres. Il s'agit de

1. http ://www.w3.org/2004/03/trix/
2. http://www.w3.org/TR/rdf-sparql-query/#rdfDataset

bases de connaissances qui sont soit construites ou validées par un expert, soit issues de données structurées (e.g. bases de données relationnelles ou documents XML).

2. Un deuxième niveau caractérise les connaissances produites par des procédés d'extraction ou d'annotation (semi-)automatiques dont le degré de fiabilité est moindre. Par exemple, les bases de connaissance DBpedia et Yago ont été construites en utilisant un processus automatique qui exploite la régularité de structuration des infobox Wikipedia. La précision des faits de Yago a été estimée à 95% ([Suchanek *et al.*, 2008]).

Deuxième technique de pondération. La deuxième technique est dédiée à la pondération des connaissances générées par notre méthode d'enrichissement. Les poids associés à ces connaissances sont calculées automatiquement au moment de leur génération. Cette technique de pondération est détaillée dans la section 6.4.2 décrivant l'approche d'enrichissement.

Dans l'exemple présenté dans la figure 6.7, l'expert a associé un poids de 1 à la base de connaissances sûre de KIM et un poids de 0.9 à la base de connaissances issue de l'annotation (c.f. figure 6.7). Un poids de 0.33 a été associé automatiquement à un extrait de la base d'enrichissement en utilisant la deuxième technique de pondération.

```
@prefix graphs: <http://reisa.com/graphs/>
@prefix sim: <http://reisa.com/sim/>

graphs:knowledgebase.web.0 {
kimkb:Laos.0 rdf:type onto:Country
...
}

graphs:knowledgebase.annotation.0 {
kimkb:Laos.0 rdf:type onto:Country
...
}
graphs:knowledgebase.enrichment.dbpedia.country.2 {
kimkb:Mekong.0 dbpedia:country kimkb:Laos.0
...
}
graphs:knowledgebase.web.0 sim:gweight 1

graphs:knowledgebase.annotation.0 sim:gweight 0.9

graphs:knowledgebase.enrichment.country.2 sim:gweight 0.33
```

FIGURE 6.7 – Exemples de graphes nommés avec leur pondération (notation TriX)

6.4 Enrichissement

Dans cette section, nous présentons notre approche d'enrichissement pour la découverte d'instances de relations sémantiques. Cette approche produit de nouvelles instances de relations sémantiques et leur associe une mesure de confiance en exploitant pour cela trois éléments : (i) la structure des documents (ii) les bases de connaissances préexistantes (BCP) et celles issues de l'annotation (BCA) et (iii) la base d'annotations.

Nous faisons l'hypothèse que plus les parties de documents annotées sont proches dans un document, plus les instances de concept associées à ces parties de document sont susceptibles d'être liées sémantiquement. Ceci justifie le fait que nous exploitons la structure des documents pour trouver des instances de relations sémantiques et leur affecter un poids.

Cette découverte d'instances de relations sémantiques est effectuée en cinq étapes comme le schématise la figure 6.8 : (1) l'identification d'instances de relations voisines sémantiquement, (2) Contrôle par la fonctionnalité des propriété et la redondance des instances de relations, (3) Filtrage par les règles du domaine, (4) Filtrage par la recherche Web et (5) saturation de la base d'enrichissement par les axiomes *rdfs:subPropertyOf*.

Nous allons présenter successivement les techniques mises en œuvre pour réaliser ces deux étapes. Cette présentation utilise les notations de la table 6.1.

Prédicat	Équivalent OWL
domain(P,C)	<P rdfs:domain C>
range(P,C)	<P rdfs:range C>
subClass(C_1,C_2)	$< C_1$ rdfs:subClassOf $C_2 >$
subProperty(P_1,P_2)	$< P_1$ rdfs:subPropertyOf $P_2 >$
fn(P)	$< P$ rdf:type owl:FunctionalProperty>
ifn(P)	$< P$ rdf:type owl:InverseFunctionalProperty>
equivalentConcept(C_1,C_2)	$< C_1$ owl:EquivalentClass $C_2 >$
equivalentProperty(P_1,P_2)	$< C_1$ owl:EquivalentProperty $C_2 >$
$refersTo(e, i)$	$< e$ sim:refersTo $i >$
$datatype(e, DT)$	$< e$ sim:datatype $DT >$
$text(e, l)$	$< e$ sim:text $l >$
$type(i, C)$	$< i$ rdf:type $C >$

TABLE 6.1 – Notations sous forme de prédicats et équivalents OWL

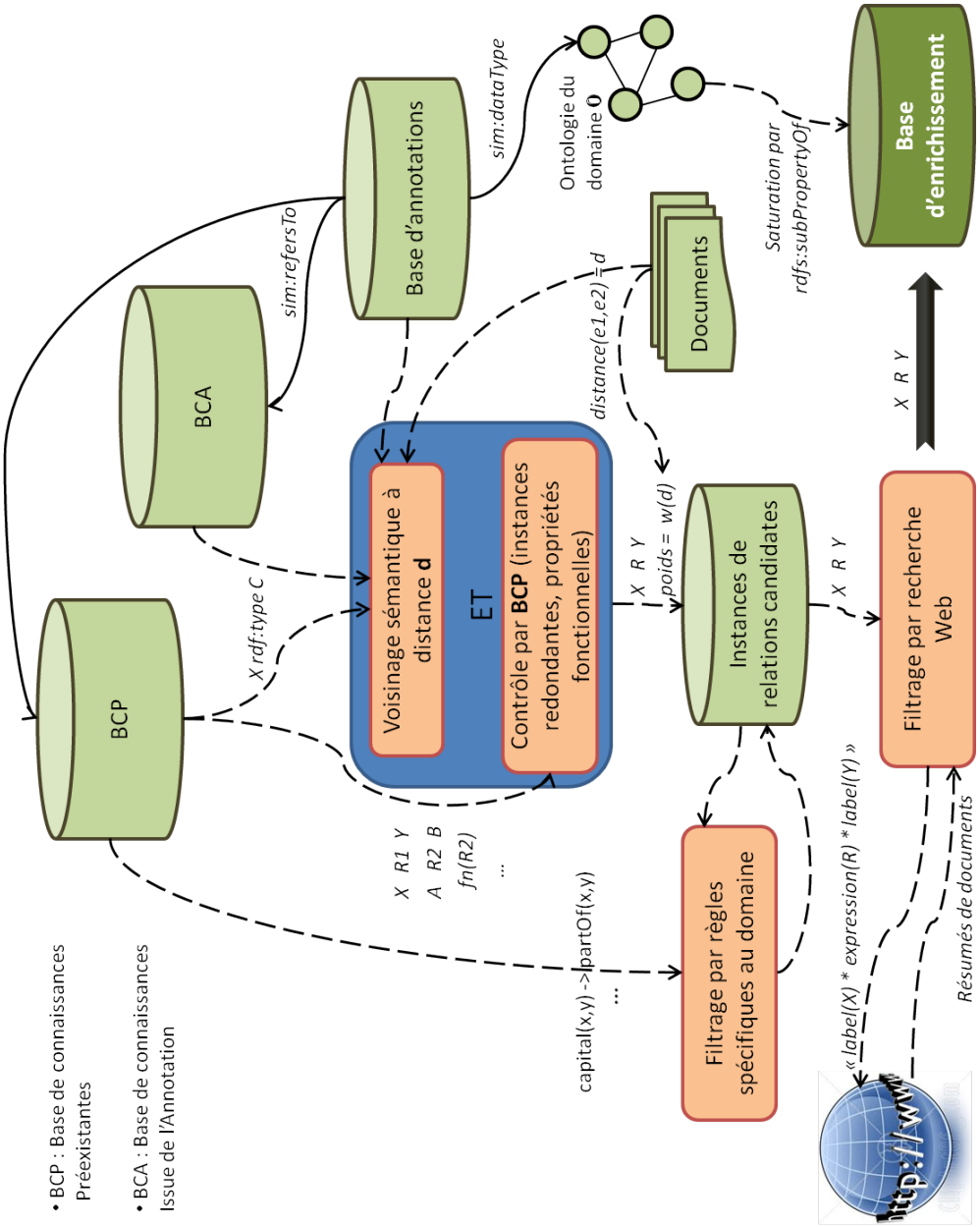

FIGURE 6.8 – Étapes du processus d'enrichissement

6.4.1 Identification des instances de relations candidates

$P(i_1, i_2)$ est considérée comme une instance de relation candidate si les instances de concepts du domaine i_1 et i_2 sont sémantiquement voisines à une distance d inférieure à un seuil donné par l'expert. La définition d'instances sémantiquement voisines est basée sur la notion de distance entre entités de document. Nous définissons successivement ces deux notions.

Distance entre entités de document

La distance entre deux entités de documents e1, e2, notée distance(e1,e2), est définie de la manière suivante :

- La distance entre deux nœuds de document e1 et e2 est la longueur du plus court chemin entre les deux nœuds dans l'arbre DOM du document.
- La distance entre deux termes $t1$ et $t2$ est la distance entre les deux nœuds les plus profonds de l'arbre DOM qui contiennent les deux termes.
- La distance entre un terme $t1$ et un nœud $n1$ est la distance entre le nœud le plus profond de l'arbre DOM qui contient le terme $t1$ et le nœud $n1$.

Voisinage sémantique à distance d

Deux instances de concept i_1 et i_2 sont dites sémantiquement voisines à distance d pour une relation P donnée, noté $V_P^d(i_1, i_2)$, ssi il existe deux entités de documents, e_1 et e_2, à distance d référant respectivement à i_1 et i_2 et que un des concepts typant i_1 est un domaine de P et un des concepts typant i_2 est un co-domaine de P. Nous formalisons cette définition de la façon suivante :

$$\begin{aligned} refersTo(e_1, i_1) \wedge refersTo(e_2, i_2) \wedge distance(e_1, e_2) = d \wedge \\ type(i_1, C_1) \wedge type(i_2, C_2) \wedge domain(P, C_1) \wedge range(P, C_2) \\ \Leftrightarrow V_P^d(i_1, i_2) \end{aligned}$$

De même, une instance de concept i et une valeur littérale l sont dites sémantiquement voisines à distance d pour un attribut P donné, noté $V_P^d(i, l)$, ssi il existe deux entités de documents, e_1 et e_2, à distance d référant respectivement à i et l, tels que l'un des concepts de l'instance i est un domaine de P et le type du littéral l est un co-domaine de P. Nous formalisons cette définition de la façon suivante :

$$\begin{aligned} refersTo(e_1, i_1) \wedge datatype(e_2, DT) \wedge distance(e_1, e_2) = d \wedge \\ type(i_1, C_1) \wedge domain(P, C_1) \wedge range(P, DT) \end{aligned}$$

$$\Leftrightarrow V_P^d(i_1, l)$$

Ainsi, déterminer le voisinage sémantique nécessite de connaître à la fois les concepts des instances référencées par les entités de document, les domaines et les co-domaines des relations. Cette dernière information est donnée par les ontologies associées aux différentes bases de connaissances exploitées.

Par exemple, dans la figure 6.9, la relation *dbpedia*:*country* entre l'instance *kimkb*:*Mekong.0* et l'instance *kimkb*:*Laos.0* est une instance de relation candidate. Ces deux instances sont référencées par deux entités de document proches (cf. figure 6.4) qui sont deux entités nommées situés dans le même nœud de document. Les instances *kimkb*:*Mekong.0* et *kimkb*:*Laos.0* sont donc voisines sémantiquement avec une distance de 0. Ces instances font parties de la base de connaissances de KIM dans laquelle se trouvent également représentés leurs types : *kim*:*River* et *kim*:*Country*, ces types étant respectivement domaine et co-domaine de la relation *dbpedia*:*country*. REISA a pu lier des instances de l'ontologie de KIM par une relation de l'ontologie de DBpedia grâce aux alignements effectués, d'une part, entre les concepts *kim*:*Country* et *dbpedia*:*Country*, et d'autre part, entre les concepts *kim*:*River* et *dbpedia*:*River*.

```
@prefix graphs: <http://reisa.com/graphs/>
graphs:candidates.dbpedia.country.0 {
kimkb:Mekong.0 dbpedia:country kimkb:Laos.0
}
graphs:candidates.dbpedia.country.1 {
kimkb:Mekong.0 dbpedia:country kimkb:Vietnam.0
}
graphs:candidates.kim.capital.2 {
kimkb:Hanoi.0 kim:capital kimkb:Vietnam.0
}
graphs:candidates.dbpedia.country.0 sim:gweight 0.8
graphs:candidates.dbpedia.country.1 sim:gweight 0.7
graphs:candidates.kim.capital.2 sim:gweight 0.6
```

FIGURE 6.9 – Exemple de triplets d'enrichissement

A la fin de cette phase, nous disposons d'instances de relations candidates entre i_1 et i_2. Ceci ne préjuge en aucun cas de la validité de cette relation entre i_1 et i_2. L'étape suivante de construction de la base d'enrichissement affine cette étape d'identification en éliminant certains des candidats.

6.4.2 Construction de la base d'enrichissement

La construction de la base d'enrichissement consiste à filtrer et pondérer les instances de relations candidates préalablement identifiées. Le filtrage se fait par l'exploitation des bases de connaissances et du Web. Dans cette section, nous commençons par présenter le rôle des bases de connaissances dans le contrôle de l'enrichissement puis celui du Web.

6.4.2.1 Contrôle par les bases de connaissances

D'une part, le but est d'éviter de générer des instances de relation déjà représentées dans les bases de connaissances avec une meilleure mesure de confiance. D'autre part, nous souhaitons ne générer que des instances de relations qui satisfont les axiomes définis dans la partie ontologique des bases de connaissances. Deux types de règles sont considérés :

(1) La fonctionnalité et la fonctionnalité inverse des propriétés. Par exemple, si la propriété *capitale* est définie comme étant une propriété fonctionnelle, et que le triplet <France,capitale,Paris> est répertorié dans le base de connaissance, le triplet candidat <France,capitale, Lyon> ne sera pas retenu.

(2) Les règles spécifiques au domaine : par exemple, si une règle de domaine précise que la date du décès d'une personne doit être ultérieure à sa date de naissance, le triplet candidat indiquant qu'une personne est née en 1990 n'est pas retenu si l'on sait, dans la base de connaissances, que cette personne est décédée en 1989.

La base d'enrichissement est ainsi construite en utilisant la base d'annotations et les bases de connaissances tout en exploitant leur degré de confiance. Comme expliqué dans la section 6.3, la pondération des faits est représentée via trois catégories de graphes nommés RDF :

1. les graphes nommés de poids 1, considérés comme des références car ils indexent des faits provenant de bases de connaissances sûres.
2. les graphes nommés indexant les faits des bases de connaissances issues de l'annotation BC_a (un poids est attribué globalement par un expert aux annotations générées par un outil donné pendant l'intégration)
3. les graphes nommés de voisinage avec un poids inférieur à 1, calculé automatiquement pendant l'enrichissement

Nous notons G^p le graphe nommé de poids p.

Graphe nommé de voisinage. Un graphe nommé de voisinage, noté $G^{w(d)}$, regroupe les faits produits par enrichissement à partir d'entités de document distantes de d dans le document. Nous définissons le poids $w(d)$ d'un graphe nommé de voisinage de manière à ce qu'il soit inversement proportionnel à la distance d. Il est calculé de la façon suivante :

$$w(d) = \frac{\alpha}{d+1} \tag{6.1}$$

où α est une constante strictement inférieure à 1 fixée par l'expert. Elle permet de prendre en compte le fait que la relation entre le poids et la distance est basée sur une heuristique. Ainsi, si deux entités de documents ont une forte proximité ($d = 0$) le poids affecté au graphe nommé de voisinage $G^{w(d)}$ ne sera pas de 1.

Cette fonction de pondération permet de trier en premier les instances de relation qui sont issues d'entités de documents à distance 0, ensuite à distance 1, etc. L'heuristique de base étant que plus les instances sont proches structurellement, plus le poids affecté à leur relation ($w(d)$) est grand. Par exemple, pour $\alpha = 0.9$, les graphes construits avec une distance de 0 auront un poids de 0.9, ceux construits avec une distance de 1 auront un poids de 0.45, ceux construits avec une distance de 2 auront un poids de 0.3, etc.

Décrivons maintenant la méthode utilisée pour construire la base d'enrichissement exploitant les graphes nommés. On part d'un ensemble de couples d'instances de concept voisines sémantiquement à distance d (E_1=$\{(i_1,i_2)$ / V_P^d (i_1,i_2) $\}$) et d'un ensemble de couples constitués d'une instance de concept et d'un littéral sémantiquement voisin à distance d (E_2=$\{(i_1,l_1)$ / V_P^d $(i_1,l_1)\}$). Les traitements appliqués aux deux ensembles sont différents. Nous les présentons successivement.

Premier cas. Une instance de propriété candidate $P(i_1, i_2)$ entre deux instances de concept i_1 et i_2 telles que $(i_1,i_2) \in E_1$, est ajoutée dans la base d'enrichissement et est indexée par le graphe $G^{w(d)}$ ssi les conditions suivantes sont satisfaites :

Condition à tester	Explication
$\forall P \neg\exists P(i_1, i_2) \in G^p$ tq. $p > w(d)$	Ce fait n'existe pas dans les graphes de meilleur poids
$fn(P) \wedge \neg\exists z$ tq. $P(i_1, z) \in G^1$	La propriété étant fonctionnelle, si i_1 est déjà liée par P à une instance de concept z dans la base de connaissances sûre, alors soit z est une référence au même objet du monde réel et il ne s'agit donc pas d'une nouvelle connaissance, soit il s'agit d'un fait candidat erroné
$ifn(P) \wedge \neg\exists\ z$ tq. $P(z, i_2) \in G^1$	raisonnement similaire pour les propriétés inverse-fonctionnelles
Il n'existe pas de règle du domaine dont l'instantiation par i_1 et i_2 conclut $\neg P(i_1, i_2)$	$P(i_1, i_2)$ est impossible compte tenu des faits de la base de connaissances préexistantes décrivant i_1 et i_2 et d'une ou plusieurs règles de domaine

Deuxième cas. Une instance de propriété $P(i,l)$ entre une instance de concept i et un littéral l est ajoutée à la base d'enrichissement et indexée par le graphe $G^{w(d)}$ ssi :

- V_P^d (i,l)
- $\neg\exists$ P$(i,l) \in G^p$ tq. $p > w(d)$
- $fn(P) \wedge \neg\exists\ m$ tq. P$(i,m) \in G^1$
- Il n'existe pas de règle du domaine dont l'instanciation par i et l conclut $\neg P(i, l)$

6.4.2.2 Contrôle par le Web

Le contrôle avec les axiomes et les instances des bases de connaissances préexistantes permet d'améliorer la précision. Cependant, l'incomplétude de ces bases de connaissances fait qu'il est toujours possible d'avoir de fausses instances de relations qui ne sont pas filtrées. Ceci est principalement dû au critère de voisinage entre les entités de documents qui ne garantit pas qu'une relation existe entre les instances de concepts référencées.

Nous nous sommes donc posés la question suivante : « Comment savoir si l'instance de relation découverte est correcte sans pour autant rechercher des régularités d'expression ou de structuration dans les documents ? ». En effet, comme nous l'avons déjà précisé dans les chapitres précédents, une relation peut avoir plusieurs formes d'expressions différentes. Aussi, nous ne pouvons pas assurer de déterminer de manière exhaustive toutes ces formes d'expression.

Le Web permet de passer outre cette limite. L'idée est que si une expression commune est identifiée pour une relation donnée, il est fort probable que cette expression soit utilisée dans un ou des documents Web pour formuler la relation. Si on considère le Web comme un corpus, c'est effectivement sa taille gigantesque et son évolution constante qui font que rechercher une relation entre deux instances

avec une ou plusieurs expressions peut être une heuristique efficace.

Ainsi, l'utilisateur de REISA peut faire le choix de contrôler les triplets d'instances de relation candidates en soumettant une expression textuelle les représentant au Web. Cette expression textuelle est construite avec un patron textuel de la relation. Par exemple, le patron "#argument1# is the capital of #argument2#" est un patron possible pour la relation *dbpedia:capital*. En remplaçant les arguments indiqués dans le patron par les labels des instances de concepts on obtient une expression textuelle pour tout le triplet candidat (e.g. "#Vientiane# is the capital of #France#").

Ces expressions textuelles sont ensuite soumises à un moteur de recherche Web (e.g. Google, Yahoo). Nous prenons en considération les résumés des dix premières réponses retournées par le moteur de recherche et nous calculons le nombre de résumés contenant exactement l'expression textuelle recherchée. Si le nombre ou la proportion de résumés qui vérifie l'expression est inférieur à un seuil fixé, nous considérons l'instance de relation comme peu fiable et elle est éliminée de la base d'enrichissement.

6.4.2.3 Saturation de la base d'enrichissement

A la fin de ce processus de filtrage, on sature la base d'enrichissement en exploitant les axiomes *rdfs:subPropertyOf* définis dans les ontologies des bases de connaissances considérées. Si une sous-propriété de P lie une instance de concept x et une instance ou un littéral y dans $G^{w(d)}$, $P(x, y)$ est ajouté à $G^{w(d)}$ si elle n'existe pas déjà dans un graphe de meilleur poids :

- $\neg\exists$ P(x,y) $\in G^p$ tq. $p > w(d)$
- $fn(P) \wedge \neg\exists\ z$ tq. P(x,z) $\in G^1$
- $ifn(P) \wedge \neg\exists\ z$ tq. P(z,y) $\in G^1$ (si y n'st pas un littéral)
- $\exists\ P'$ tq. $subProperty(P', P)$ et $P'(x,y) \in G^{w(d)}$
  Les instances de propriétés sont propagées avec le même poids. La probabilité d'appartenance à un ensemble est en effet au moins aussi grande que la probabilité d'appartenance à un sur-ensemble [Zadeh, 1965].

6.4.2.4 Algorithme de construction de la base d'enrichissement

La figure 6.8 résume les différentes étapes de l'algorithme de construction de la base d'enrichissement détaillées ci-dessus. Les instances de relations candidates sont construites en commençant par les distances de voisinage les plus courtes. L'algorithme est appliqué jusqu'à un seuil de distance μ fixé par l'expert. Nous

rappelons qu'un graphe nommé $G^{w(d)}$ est créé pour chaque distance d. Le poids du graphe, w(d), est indiqué avec le triplet ($G^{w(d)}$, sim:gweight, $w(d)$).

L'exemple décrit dans la figure 6.9 montre les graphes de voisinage résultant du module d'enrichissement appliqué à la base d'annotations et à l'extrait de la base de connaissances WKB de l'exemple précédent (cf. figure 6.4). Les poids ont été calculés avec α=0.9 et un seuil de distance μ=3 (distances structurelles 0, 1 et 2).

REISA a pu inférer ici trois relations candidates à partir des annotations de l'extrait de document et de la base de connaissances disponible (cf. figures 6.2 et 6.4) :

- La relation *dbpedia :country* entre la rivière "Mékong" et le pays "Laos" avec une distance de voisinage de 0.
- La relation *dbpedia :country* entre la rivière "Mékong" et le pays "Vietnam" avec une distance de voisinage de 1.
- La relation *kim:capital* entre la ville "Hanoi" et le pays "Vietnam" avec une distance de voisinage de 2.

La relation *dbpedia:country* telle qu'exprimée dans l'ontologie DBpedia permet d'indiquer qu'une rivière donnée traverse un pays donné.

La propriété *kim:capital* a pu être inférée grâce aux entités nommées "Hanoi" et "Vietnam" qui étaient à distance 2 dans le document mais non pour "Vientiane" et "Vietnam" qui étaient à distance 0 dans le document. La fonctionnalité de la propriété *dbpedia :capital* nous a permis d'éviter d'associer la ville *Vientiane* comme étant capitale du *Vietnam*, puisque la ville est déjà connue comme capitale d'un autre pays (i.e. le *Laos*) dans la base de connaissances. Ainsi, nous avons pu retrouver la bonne capitale du *Vietnam*. Le filtre par le Web a permis de confirmer cette relation en retournant 8 résumés contenant l'expression "Hanoi * capital of * Vietnam" parmi les dix premières réponses.

6.5 Interrogation

Dans la phase d'interrogation, notre approche permet de répondre à des requêtes SPARQL exprimées à l'aide du vocabulaire d'une ontologie de domaine en les réécrivant pour cibler les graphes nommés concernés.

L'objectif de la réécriture est d'exploiter les graphes nommés construits pour (i) atteindre de nouvelles réponses grâce aux nouvelles instances de relations de la base d'enrichissement et à leur combinaison avec les relations définies dans les

bases de connaissances préexistantes et (ii) trier les réponses selon leur degré de confiance en utilisant les poids affectés aux graphes nommés les indexant.

Le processus de réécriture s'applique aux requêtes SPARQL exprimées suivant le vocabulaire d'une ontologie de domaine. Pour une requête $Q(P, S, F, D)$ donnée, la reformulation consiste à :

1. Rechercher les graphes nommés $?g_i$ auxquels appartiennent les patrons de triplet t_i de la requête utilisateur. Plus précisément, la réécriture syntaxique utilise le mot-clé SPARQL "GRAPH" en substituant chaque patron de triplet $t_i \in P$ par $GRAPH\ ?g_i\ \{t_i\}$. Cela retourne l'IRI des graphes indexant t_i dans la variable $?g_i$
2. Récupérer les poids $?p_i$ des graphes $?g_i$ grâce à la propriété $sim : gweight$
3. Demander le tri des réponses suivant une fonction d'agrégation sur l'ensemble des poids. Cela se fait par la primitive SPARQL $ORDER\ BY$. Plusieurs fonctions d'agrégation peuvent être utilisées comme les fonctions maximum, moyenne ou minimum.

La figure 6.10 présente un exemple de cette réécriture pour une requête en entrée demandant la liste des pays avec leur capitale.

	Requête initiale	Requête réecrite
1)	SELECT ?c ?t WHERE {	SELECT ?c ?t WHERE {
2)	?c *rdf* : *type kim* : *Country*	GRAPH ?g1 { ?c *rdf* : *type kim* : *Country* }
3)	?t *rdf* : *type kim* : *City*	GRAPH ?g2 { ?c *rdf* : *type kim* : *City* }
4)	?t *kim* : *capital* ?c	GRAPH ?g3 { ?t *kim* : *capital* ?c }
5)	}	?g1 *sim* : *gweight* ?p1
6)		?g2 *sim* : *gweight* ?p2
7)		?g3 *sim* : *gweight* ?p3
8)		} ORDER BY avg(?p1, ?p2, ?p3)

FIGURE 6.10 – Exemple de reformulation d'une requête SPARQL

Dans cet exemple de réécriture la fonction *avg* (moyenne) est utilisée pour trier les réponses suivant leur poids. Les lignes 2, 3 et 4 correspondent à l'étape 1 du processus de reformulation, les lignes 5,6,7 à l'étape 2 et la ligne 8 correspond à l'étape 3.

6.6 Conclusion

Dans ce chapitre nous avons présenté notre approche d'enrichissement automatique contrôlé par des bases de connaissances. Cette approche utilise les faits et

les axiomes présents dans les bases de connaissances préexistantes pour filtrer les instances de relations candidates. Elle exploite aussi les appels Web comme un filtre statistique supplémentaire qui permet d'augmenter la précision des instances de relations.

Dans le chapitre suivant nous présentons l'évaluation de l'approche REISA sur deux corpus réels extraits du Web ayant des caractéristiques très différentes. Nous évaluons l'efficacité des différents contrôles (filtres) de manière séparée. La précision des instances de relations découvertes est mesurée en appliquant les filtres progressivement et en évaluant les résultats intermédiaires obtenus après chaque filtre.

Chapitre 7

Évaluation et synthèse de l'approche REISA

Dans ce chapitre, nous présentons les évaluations de notre approche d'enrichissement. L'objectif des expérimentations effectuées est d'évaluer la précision des instances de relations incertaines générées par REISA en fonction du poids des graphes nommés auxquels elles appartiennent et de l'utilisation des différents contrôles effectués grâce aux bases de connaissances ou au Web.

Les expérimentations ont été menées sur deux domaines et deux corpus différents : un premier corpus extrait du Web relatif aux appels à communications pour événements scientifiques et un deuxième corpus extrait de Wikipedia relatif

aux entités géographiques. Ces deux couples domaine-corpus ont des caractéristiques différentes. Le premier est peu hétérogène sémantiquement : l'ontologie n'exprime qu'une relation au maximum entre deux concepts donnés et les instances de concept mentionnées dans le corpus sont typées par peu de concepts de l'ontologie (e.g. Personne, Évenement, Ville). Le deuxième couple domaine-corpus est plus hétérogène sémantiquement : l'ontologie exprime plusieurs relations possibles entre deux concepts donnés et les instances de concepts mentionnées dans le corpus sont typées par un ensemble très varié de concepts.

Dans un premier temps, nous appliquons notre approche d'enrichissement sur ces deux corpus avec des bases de connaissances préexistantes différentes. Dans un second temps, nous évaluons les contenus des graphes nommés construits par enrichissement selon les heuristiques définies pour leur construction. Nous n'évaluons pas la partie interrogation de l'approche REISA car la qualité des réponses obtenues dépend uniquement de la qualité des triplets générés par enrichissement. Par ailleurs, la combinaison des faits issus de l'enrichissement avec des faits provenant des bases préexistantes sûres dans les réponses ne permet pas de voir clairement l'apport de l'enrichissement. Dans ce qui suit, nous présentons les deux expérimentations effectuées pour l'évaluation de l'approche sur les deux corpus.

7.1 Première expérimentation

Dans cette première expérimentation, nous nous intéressons au domaine des appels à communication et à un premier type de corpus/domaine restreint sémantiquement : l'ontologie n'exprime qu'une relation au plus pour un couple de concepts et le corpus réfère à des instances typées par un ensemble restreint des concepts de l'ontologie. Dans cette section, nous commençons par présenter le corpus et l'ontologie de référence utilisés. Nous présentons ensuite les bases de connaissances préexistantes que nous exploitons et les méthodes utilisées pour annoter le corpus. Enfin, nous discutons les résultats d'évaluation de notre approche d'enrichissement appliquée à l'ensemble de ces éléments.

7.1.1 Corpus

Ce premier corpus est constitué de 511 pages Web téléchargées de 32 sites d'appels à communication en langue anglaise. Ce corpus comporte des entités nommées de différents types. Il décrit des événements scientifiques, les différentes dates qui leur sont associées, leurs lieux et les membres des comités de programme et des comités d'organisation.

7.1.2 Base de connaissances préexistante

Nous avons formé une base de connaissances "sûre", BC_1, composée d'extraits de la base de connaissances DBLP-RDF [1] qui décrit des références bibliographiques et d'extraits de la base de connaissances WKB de KIM [2] qui est généraliste. DBLP-RDF utilise les ontologies *SWRC* et *FOAF* comme référence. *WKB* utilise l'ontologie *PROTON* (dont le namespace est préfixé par kim dans ce qui suit). Dans nos expérimentations, nous avons utilisé l'union de ces deux ontologies [3].

Plus précisément, l'ontologie de BC_1 (cf. figure 7.1) représente des articles avec le concept *swrc*:*InProceedings*, des auteurs avec le concept *foaf*:*Agent* et son équivalent *kim*:*Person*, des actes avec le concept *swrc*:*Proceedings*, des séries avec le concept *swrc*:*Conference* (e.g. ESWC), des conférences avec le concept *reisa*:*Event* que nous avons ajouté (e.g. ESWC 2010) et pour chaque conférence, le pays avec le concept *kim*:*Country*, la ville avec le concept *kim*:*City* et de manière générale son lieu avec le concept *kim*:*Location*.

Pour représenter le lien entre les conférences et leurs lieux, nous avons défini la propriété *reisa*:*hasLocation* et ses sous-propriétés fonctionnelles *reisa*:*hasCountry* et *reisa*:*hasCity*. Le titre de la conférence est sauvegardé avec l'attribut *reisa*:*title* et son année avec l'attribut *reisa*:*year*.

Les faits de la base de connaissances BC_1 sont les suivants :

- les actes des événements qui se sont déroulés entre 2003 et 2007 de DBLP-RDF : 5608 instances de *swrc*:*Proceedings* et leur description,
- les villes, pays et localisations décrits dans *WKB* et leur description : 12484 instances de *kim*:*Location*, 3093 de *kim*:*City*, 502 de *kim*:*Country*,
- les instances des évènements *reisa*:*Event* correspondant aux actes (*swrc*:*Proceedings*) et décrites par une date (la date du proceeding), un titre (le titre de la série concernée) et éventuellement une ville et un pays, quand l'information est présente dans le titre DBLP de la conférence. Sur les 5608 instances de *reisa*:*Event*, la ville est inconnue pour 3668 (65.40%) d'entre elles, le pays est inconnu pour 700 instances (12.48%) et 518 (9.23%) des instances ne sont rattachées ni à une ville ni à un pays.

1. http://thedatahub.org/dataset/fu-berlin-dblp
2. http://www.ontotext.com/kim/semantic-annotation
3. http://www.lri.fr/~mrabet/reisa-onto.rdf

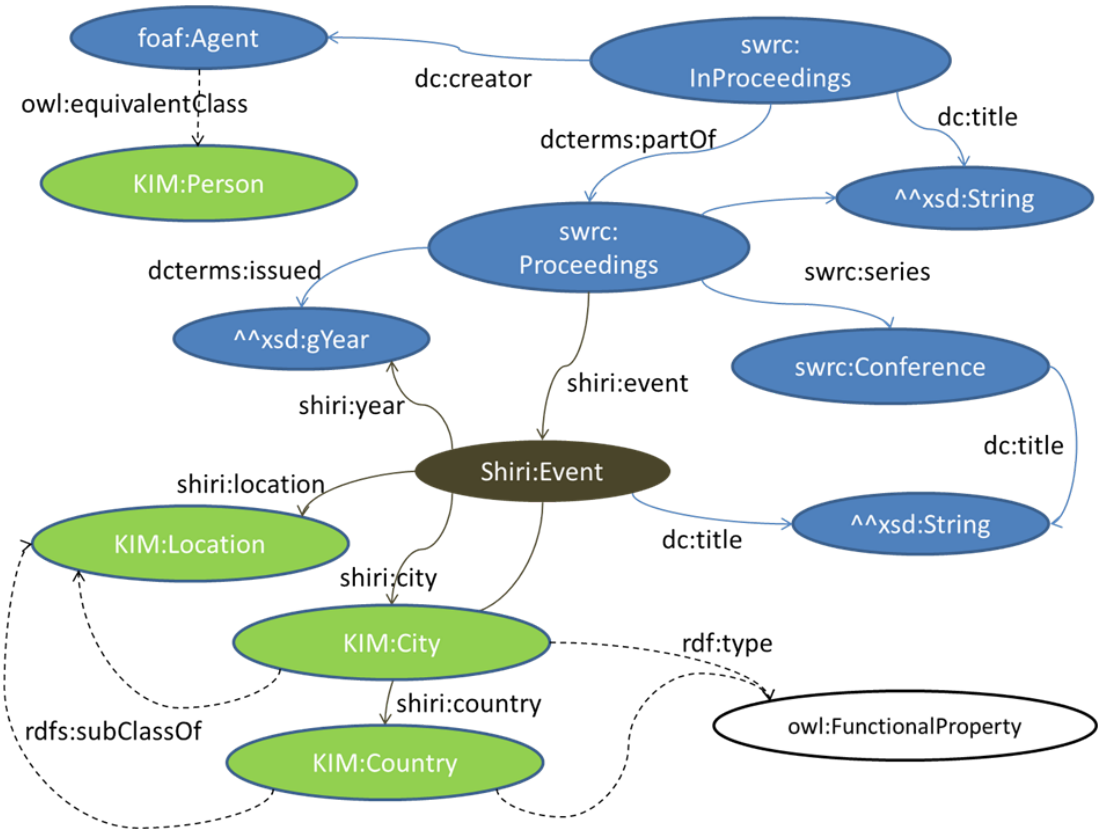

FIGURE 7.1 – Ontologie de référence

7.1.3 Annotation du corpus

L'annotation du corpus exploite aussi bien l'ontologie de référence que les bases de connaissances préexistantes.

L'ensemble des documents est annoté en utilisant des outils existants ou des utilitaires que nous avons développés. Cette annotation permet de générer les triplets *sim:refersTo* permettant de relier une entité de document à une instance existante dans les bases de connaissances ou à de nouvelles instances stockées dans la base de connaissances issue de l'annotation BCA. Les entités de documents sont annotées au niveau terme. Une entité de document est ainsi identifiée par une URI correspondant à la concaténation de l'URL du document, de chemin XPath du nœud parent du terme et de la position du terme dans le nœud. Par exemple, l'identifiant "corpus/doc0/html/body/div/p[3]/a.0" caractérise le terme à la position 0 dans le nœud "/html/body/div/p[3]/a" du document "doc0" du corpus exploité. Le contenu textuel de l'entité de document est ici le terme lui même.

Comme les informations concernant les lieux des conférences sont incomplètes dans la base de connaissances BC_1, nous nous sommes intéressés à l'annotation des lieux et des événements. Nous avons utilisé pour cela deux procédés. Le premier exploite la plateforme d'annotation KIM pour retrouver les occurrences de villes, pays et autres lieux qui sont soit reconnues comme référant à des instances de la BC de KIM, soit reconnues comme nouvelles instances. Cette annotation a permis de retrouver 163 instances de villes, 203 instances de pays et 2 instances de lieux qui ne sont ni des villes ni des pays.

Le deuxième procédé exploite des patrons lexicaux et des listes de mots vides pour retrouver des titres d'événements (e.g. conférences, ateliers) et leur date.

L'entité de document regroupant le titre et la date extraits est sauvegardée avec un lien *refersTo* vers une instance de type *shiri:Evenement* qui peut être soit :

- une instance de BC_1 si le titre et la date correspondent à un événement existant dans BC_1. Cette correspondance a été effectuée en utilisant la similarité des N-grams pour les titres longs. Des mesures spécifiques ont été définies pour le cas des acronymes.
- une instance retrouvée lors de l'annotation d'une autre entité de document (même titre et même date).
- une nouvelle instance si aucune correspondance n'a pu être retrouvée.

L'application de ce processus d'annotation a permis de retrouver 1429 entités de documents référant à des événements, 840 entités référant à des villes et 1618 entités référant à des pays.

Sur les 1429 entités de document référant à des événements, 348 réfèrent à des instances de BC_1 et 1081 entités réfèrent à des nouvelles instances (découvertes dans les documents).

Les connaissances produites à partir de l'annotation du corpus sont réunies dans une base de connaissances BC_a et indexées par le graphe nommé $G^{0.9}$ qui a un poids de 0.9. Cette valeur de poids a été attribuée pour indiquer que les triplets de ce graphe sont jugés moins sûrs que les triplets de BC_1. Les entités de document et les liens $refersTo$ ont été sauvegardés dans la base d'annotations B_A.

7.1.4 Construction de la base d'enrichissement

Nous avons construit la base d'enrichissement en appliquant l'approche REISA avec la base de connaissances BC_1 et le corpus annoté de 511 documents. Ceci a permis de retrouver 187 propriétés (instances des propriétés *reisa*:*hasCountry*, *reisa*:*hasCity* et *reisa*:*hasLocation*), avec un seuil $\mu = 2$ pour la distance de voisinage et avec α fixé à 0.9 pour le calcul du poids. Rappelons ici que le poids d'un graphe nommé est égal à $\alpha/(d+1)$, d étant la distance de voisinage qui a été utilisée pour la construction du graphe (cf. section 6.3).

7.1.5 Évaluation et discussion

Dans une première évaluation, nous mesurons la précision des triplets générés par voisinage (i.e. contenu des graphes nommés) pour les propriétés fonctionnelles *reisa*:*country*, *reisa*:*city* et *reisa*:*location*. La figure 7.2 montre la valeur de précision totale obtenue en fonction de la distance de voisinage entre les entités de documents annotées. Nous avons effectué différents tests en construisant la base d'enrichissement de quatre façons différentes pour évaluer l'apport de chaque heuristique définie dans l'approche REISA. Ces tests sont présentés dans le tableau 7.1.

Test	Méthode de construction de la base d'enrichissement
V	Voisinage sémantique uniquement
VF	V + critère de fonctionnalité
VR	V + élimination des propriétés existantes dans BC_1
VRF	VR + critère de fonctionnalité
VRFL	VRF + règles du domaine
VRFLW (REISA)	VRFL + filtrage par le Web

TABLE 7.1 – Tests définis pour l'évaluation

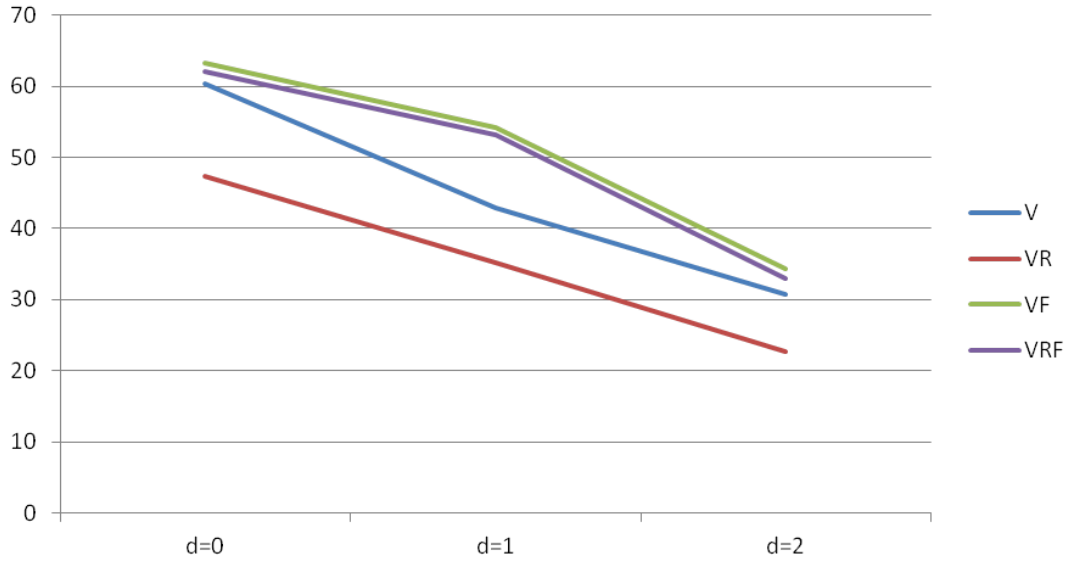

FIGURE 7.2 – Précision des faits enrichis en fonction de la distance de voisinage

	hasCity			hasCountry			hasLocation		
	d≤0	d≤1	d≤2	d≤0	d≤1	d≤2	d≤0	d≤1	d≤2
V	22	34	57	25	50	102	48	84	159
VR	19	28	46	17	42	90	38	71	137
VF	15	29	42	14	18	53	48	84	159
VRF	15	29	42	13	17	51	38	71	137

TABLE 7.2 – Nombre de triplets trouvés par configuration et distance

La précision a été calculée sur l'ensemble des relations visées par l'enrichissement suivant la formule suivante :

$$Precision = \frac{Nombre\ de\ relations\ candidates\ correctes}{Nombre\ de\ relations\ candidates} \tag{7.1}$$

Le tableau 7.2 présente le nombre de triplets trouvés par l'approche d'enrichissement par relation et par test. Le tableau 7.3 présente le nombre de triplets corrects trouvés par relation et par test. Enfin le tableau 7.4 présente la précision des faits construits par enrichissement par relation et par test.

Variation de la précision en fonction de la distance de voisinage (poids des graphes)

La qualité (précision) de ces nouveaux faits varie en fonction de la distance de

	hasCity			hasCountry			hasLocation		
	d≤0	d≤1	d≤2	d≤0	d≤1	d≤2	d≤0	d≤1	d≤2
V	10	13	16	18	22	32	29	36	49
VR	7	10	10	10	14	20	18	25	31
VF	7	10	10	11	15	22	29	36	49
VRF	7	10	10	10	14	20	18	25	31

TABLE 7.3 – Nombre de triplets corrects trouvés par configuration et distance

	hasCity			hasCountry			hasLocation		
	d≤0	d≤1	d≤2	d≤0	d≤1	d≤2	d≤0	d≤1	d≤2
V	45,45	38,23	28,07	72,00	44,00	31,37	60,41	42,85	30,81
VR	36,84	35,71	21,73	58,82	33,33	22,22	47,36	35,21	22,62
VF	46,66	34,48	23,80	78,57	83,33	41,50	60,41	42,85	30,81
VRF	46,66	34,48	23,80	76,47	77,77	39,21	47,36	35,21	22,62

TABLE 7.4 – % de précision par relation, distance de voisinage et configuration

	hasCity			hasCountry			hasLocation		
	d=0	d=1	d=2	d=0	d=1	d=2	d=0	d=1	d=2
	70,00	76,92	62,5	55,55	63,63	62,5	62,06	69,44	63,26

TABLE 7.5 – % de nouvelles réponses sur les réponses retrouvées (VRF/V)

voisinage. A distance 0 (même nœud de document), pratiquement un fait sur 2 est correct pour les villes et 76.5% des faits sont corrects pour les pays. La précision diminue notablement quand la distance augmente. Cependant, nous notons que pour la relation *hasCountry*, la précision augmente à distance 1 par rapport à la distance 0 (elle passe de 76,5% à 77,8%). Cela est dû à deux éléments : (i) de nouvelles réponses correctes sont retrouvées à distance 1 et (ii) le critère de fonctionnalité élimine plus de mauvaises réponses à distance 1 qu'à distance 0.

Variation de la précision en fonction de l'application du critère de fonctionnalité

Les résultats présentés dans le tableau 7.4 montrent que le critère de fonctionnalité a un impact important sur la précision des faits candidats. Notons que l'évaluation de cet impact se fait en comparant les résultats obtenus par les tests VR (faits candidats n'existant pas dans BC_1) et VRF (faits candidats n'existant pas dans BC_1 et filtrés par le critère de fonctionnalité).

Dans notre expérimentation, les propriétés fonctionnelles sont *hasCity* et *hasCountry*. La précision augmente pour les deux propriétés quelle que soit la distance, à l'exception de *hasCity* à distance 1. Par exemple, pour la propriété *hasCountry*, ce critère améliore la précision de 58,82% à 76,47% à distance 0.

Apport en termes de connaissances nouvelles

Nous considérons qu'un fait (sujet,prédicat,objet) est nouveau s'il n'existe pas en base de connaissances et si aucun autre fait de la base de connaissances ne relie le sujet à un autre objet si le prédicat est une relation fonctionnelle.

Le tableau 7.5 présente le pourcentage de faits nouveaux corrects générés par notre approche (test VRF) par rapport au nombre total de faits corrects trouvés (faits obtenus par le test V). Le pourcentage de faits nouveaux obtenus grâce à notre approche d'enrichissement varie entre 55 et 76 % de l'ensemble des faits trouvés pour les trois relations *hasCity*, *hasCountry* et *hasLocation*.

Évaluation de l'utilisation des règles de domaine et de la recherche Web

Pour ce premier corpus, la construction de la base d'enrichissement avec le test $VRFL$ a consisté à utiliser la règle du domaine suivante :

$$hasCountry(x, y) \wedge partOf(z, w) \wedge w <> y \Rightarrow \neg hasCity(x, z) \tag{7.2}$$

Ainsi, si une instance candidate $hasCity(i_1, i_3)$ est proposée par enrichissement

pour l'événement i_1 et que le fait $hasCountry(i_1, i_2)$ est présent dans la base de connaissances préexistante BC_1, la ville proposée par enrichissement, i_3, doit vérifier $partOf(i_3, i_2)$ pour que l'instance candidate $hasCity(i_1, i_3)$ soit retenue.

Cependant, cette règle n'a pas permis de filtrer plus de candidats erronés dans cette expérimentation. Cela est dû (i) au peu de liens de références retrouvés vers les instances de la base de connaissances préexistantes ($\tilde{2}5\%$ des liens de références) et (ii) au fait que ces instances n'avaient pas l'information *hasCountry* de disponible dans la base de connaissances.

D'autre part, l'application du Web n'a pas permis d'améliorer la précision des résultats obtenus. Cela est principalement dû au fait que les labels des instances du concept *reisa :Event* sont très longs (e.g. "Conference on Web Services, Nevada, Westin Horton Plaza San Diego" a été extrait comme un titre/label d'évènement). Ces labels font que les requêtes posées au Web avec les expressions textuelles des relations recherchées (e.g. "X is held in Y") n'ont pas ou très peu de résultats, ce qui est insuffisant pour filtrer davantage les instances de relation candidates.

7.2 Deuxième expérimentation

Dans cette deuxième expérimentation, nous nous intéressons au domaine géographique et à un deuxième type de corpus/domaine hétérogènes sémantiquement : l'ontologie exprime plusieurs relations possibles pour un même couple de concepts et le corpus réfère à des instances typées par un ensemble varié de concepts de l'ontologie. Dans cette section, nous commençons par présenter le corpus et l'ontologie de référence utilisés. Nous présentons ensuite les bases de connaissances préexistantes que nous exploitons et les méthodes utilisées pour annoter le corpus. Enfin, nous présentons et discutons les résultats d'évaluation de notre approche d'enrichissement appliquée à l'ensemble de ces éléments.

7.2.1 Corpus

Nous avons défini une procédure de collecte automatique de corpus en exploitant la base de connaissances DBpedia. Le corpus sélectionné pour cette expérimentation correspond à tous les articles Wikipedia sur les chaînes de montagnes de France, dont les URLs ont été collectées en soumettant la requête SPARQL suivante (cf. figure 7.3) au serveur DBpedia :

Nous avons ensuite téléchargé et nettoyé automatiquement les 27 documents cor-

```
PREFIX foaf : <http ://xmlns.com/foaf/0.1/>
PREFIX dbpedia : <http ://dbpedia.org/class/yago/>
SELECT ?page WHERE {
 ?instance rdf:type dbpedia:MountainRangesOfFrance.
 ?instance foaf:page ?page .
}
```

FIGURE 7.3 – Exemple de requête de sélection de corpus

respondant aux URLs obtenues avec l'outil *HTML Cleaner*[1]. Le nettoyage a consisté, entre autres, à supprimer tous les scripts, les formulaires, les balises "meta" de l'entête des documents et à enlever les balises de mise en forme (e.g. balises <font>, <U>, <sup>, <B>) tout en gardant leur contenu.

Cependant, bien que les articles aient été sélectionnés en spécifiant un type sémantique, leur contenu reste très hétérogène avec des sections parlant, par exemple, de sports, de faune, de flore, de climats, de villes ou de pays.

7.2.2 Bases de connaissances préexistantes

Dans cette expérimentation, nous avons utilisé la base de connaissances DBpedia. Plus précisément, nous avons sélectionné l'extrait de la base de connaissances qui décrit les instances de concepts présentes dans les documents du corpus. Cette extraction vise à optimiser le processus d'enrichissement qui nécessite des données de la base de connaissances et des données de la base d'annotations locale qui décrit les liens *sim:refersTo*.

Le nombre d'instances de concepts référencées dans le corpus est de 420. L'extrait de la base de connaissances que nous avons sélectionné est constitué des faits (triplets) qui ont pour sujet ou objet une de ces instances. Le nombre de triplets obtenus par cette extraction est 281,260.

7.2.3 Annotation du corpus

Nous avons utilisé les formes de surface Wikipedia pour annoter le corpus [Gerber & Ngomo, 2011]. Ces formes correspondent aux textes des hyperliens HTML pointant vers des documents Wikipedia. Puisque un grand nombre de documents Wikipedia est associé à des instances de concepts dans la base de connaissances

1. http://htmlcleaner.sourceforge.net/

DBpedia, l'annotation a consisté à considérer le texte de l'hyperlien comme une référence à l'instance de concept DBpedia associé au document cible.

Cependant, nous n'avons pas exploité tous les hyperliens quelque soit leur type. Dans un soucis de cohérence nous avons annoté uniquement les formes de surface référant à des instances des concepts DBpedia *NaturalPlace* et *PopulatedPlace*.

Ce choix rentre dans le cadre des trois catégories de relations que nous avons sélectionnées pour l'enrichissement et qui sont présentées dans le tableau 7.6.

7.2.4 Construction de la base d'enrichissement

Nous avons appliqué l'approche REISA avec différents configurations afin de construire la base d'enrichissement suivant les tests du tableau 7.1. Nous avons ciblé 16 relations de l'ontologie de DBpedia. Nous avons obtenus des faits candidats uniquement pour 6 relations : le voisinage sémantique n'a pas permis de retrouver des candidats pour les 9 autres relations avec le seuil de distance de voisinage fixé à 10 pour cette expérimentation. Parmi les 6 relations qui ont été retrouvées par voisinage, 3 n'ont pas un nombre de candidats suffisant pour permettre une évaluation significative.

Le tableau 7.6 présente les 3 relations retenues pour cette évaluation, ainsi que leurs domaine et co-domaines, les axiomes les décrivant et les règles de domaine que nous exploitons pour contrôler la production de leur instances dans REISA.

Relation : R	domain(R,C)	range(R,C)	fn(R)	Règles du domaine
mouthCountry	*River*	*Country*	✓	$type(y, River) \wedge type(x, LandLockedCountries) \Rightarrow \neg mouthCountry(y, x)$: Un pays doit être voisin d'une mer ou d'un océan
sourceCountry	*River*	*Country*	✓	
bdw-country	*BodyOfWater*	*Country*		

TABLE 7.6 – Relations évaluées

7.2.5 Évaluation et discussion

La figure 7.4 montre la précision des graphes nommés d'enrichissement obtenus suivant la configuration utilisée pour leur construction (cf. tableau 7.1).

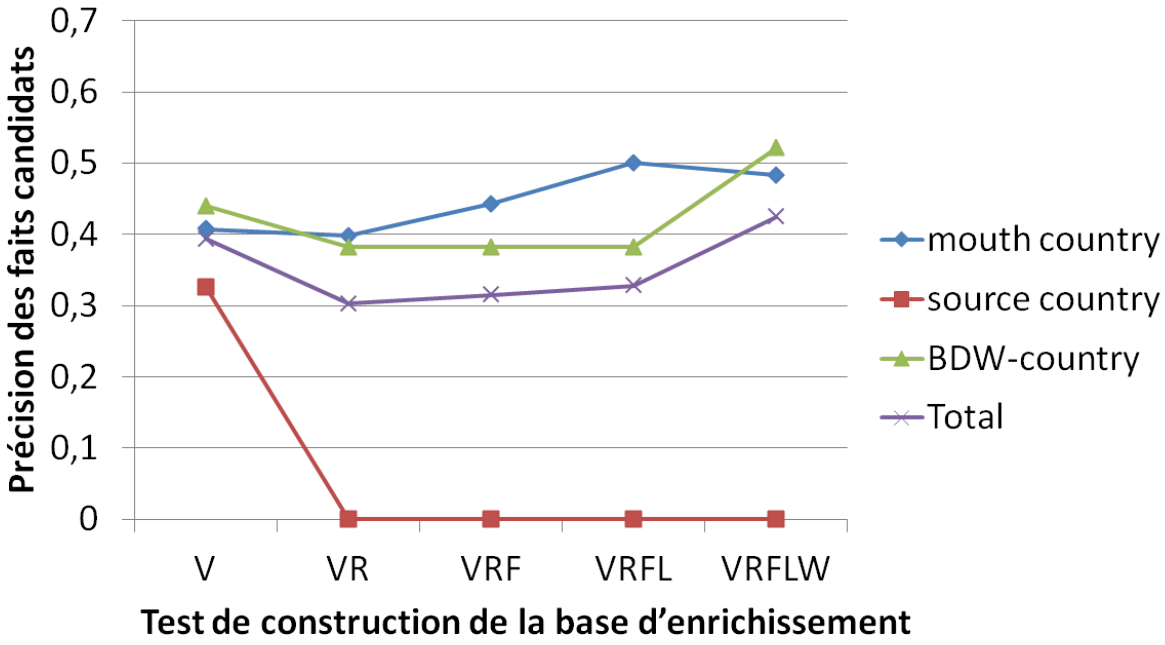

FIGURE 7.4 – Précisions des faits des graphes nommés suivant la configuration

La valeur α a été fixé à 0.9. Rappelons ici que le poids d'un graphe nommé est égal à $\alpha/(d+1)$, d étant la distance de voisinage qui a été utilisée pour la construction du graphe.

Les résultats montrent que le contrôle par la base de connaissances permet d'augmenter la précision d'une façon significative grâce à l'exploitation des axiomes de fonctionnalité. Le test V a permis de rendre compte de la pertinence relative du voisinage sémantique pour retrouver les relations sémantiques, même si celles-ci ne sont pas toujours exprimées dans les documents. Par exemple, la précision augmente de 39% à 44% pour la relation *mouthCountry* avec le test VRF. La règle de domaine utilisée a aussi permis d'augmenter la précision de 44% à 50% pour la relation *mouthCountry* avec le test $VRFL$.

Une exception est à noter cependant pour la relation *sourceCountry* où toutes les instances de relations correctes retrouvées au test V sont déjà présentes dans la base de connaissances. Dès l'application du filtre VR, toutes les bonnes instances ont été filtrées et les seuls faits retenus étaient des faits erronés. Le contrôle par le Web a cependant permis d'éliminer certains faux candidats.

D'un autre côté, le contrôle par le Web a permis d'améliorer la précision totale sur toutes les relations de 32% à 42%. Cependant, dans certains cas ce filtre a

éliminé en même temps des instances erronées et des instances correctes. Cela est le cas de la relation *mouthCountry* dont la précision a diminué de 50% à 48% avec le filtre Web car le filtre élimine plus de candidats corrects que de candidats erronés.

Cela est dû au fait que les patrons de relations exploités pour la construction d'expressions textuelles des instances sont trop restrictifs (e.g. "X * mouth * Y") pour cette relation. A cette restriction s'ajoute parfois le label des instances de concepts sujet et objet des relations candidates, qui ne représentent pas toujours la forme la plus fréquente d'expression (e.g. "Fos, Haute-Garonne" est le label anglais pour l'Haute-Garonne). Rappelons aussi que le contrôle par le Web n'est pas effectué si un nombre minimum de résumés de documents n'est pas obtenu pour les requêtes soumises. Nous avons fixé ce seuil à 6 dans nos expérimentations.

La figure 7.5 montre la précision des faits des graphes nommés suivant le seuil de distance de voisinage d.

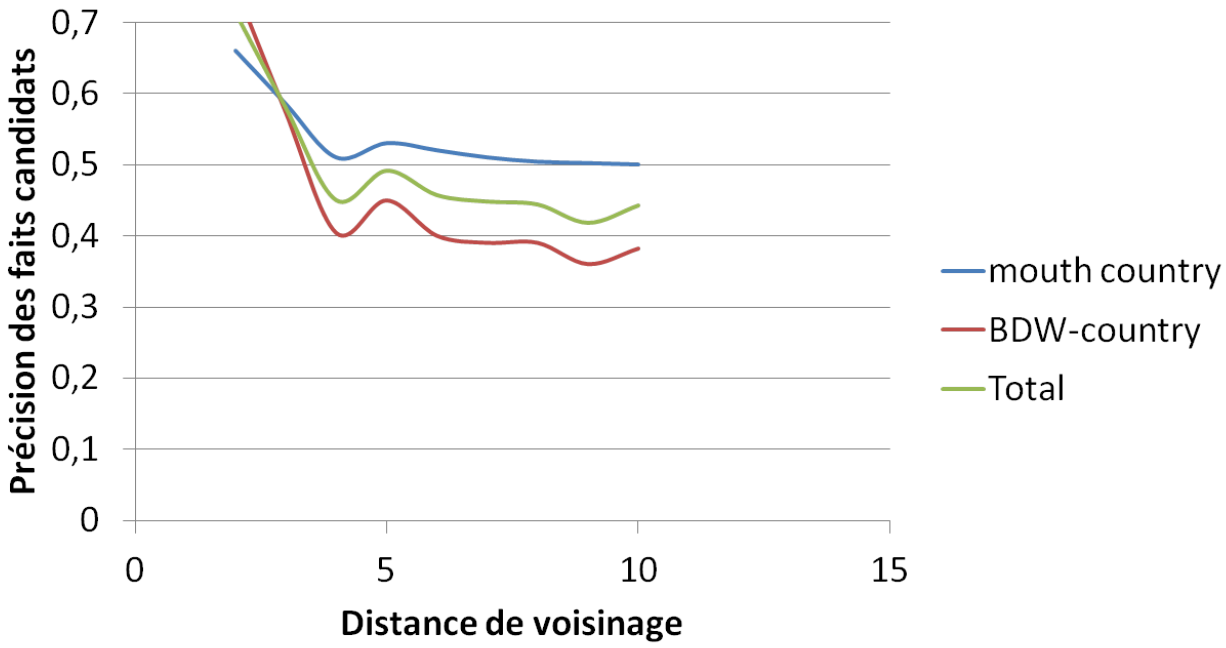

FIGURE 7.5 – Précision des faits des graphes nommés suivant la distance

La précision des faits candidats diminue quand la distance de voisinage augmente. Quelques exceptions sont à noter. Par exemple, pour le seuil de distance 5, la précision augmente de 4% par rapport à la précision obtenue avec les distances de voisinage ≤ 4. Cela est potentiellement dû à une structuration implicite de

certains documents qui fait que plus de candidats sont trouvés à distance 5 qu'à distance 4 pour l'ensemble des relations testées. Cependant, la courbe de variation de la précision par rapport au seuil de distance de voisinage (cf. figure 7.5) montre bien que les intervalles de seuil de distance sont des indicateurs de la précision des faits candidats (e.g. intervalles 2 à 5, 5 à 6 et 6 à 10).

7.3 Conclusion

Nous avons évalué l'approche REISA sur deux corpus de domaines différents. Les résultats montrent que la distance de voisinage des entités de document annotées est un bon critère pour trier les instances de relations candidates.

L'expérimentation sur les deux corpus a permis d'étudier l'intérêt de l'exploitation des bases de connaissances pour le contrôle de l'enrichissement. Les effets du critère de fonctionnalité sur la précision des faits construits sont visibles sur les deux corpus. Les règles de domaine permettent également d'améliorer la précision. Ces deux types de contrôle sont d'autant plus efficaces si suffisamment de liens sont découverts entre les entités de document et les instances de concepts des bases de connaissances préexistantes.

Le contrôle par le Web permet aussi d'améliorer la précision si les patrons de relations exploités sont d'utilisation courante sur le Web et si les labels des instances de concepts des bases de connaissances sont représentatives de l'instance de concept.

D'autres expérimentations sont aussi nécessaires pour étudier plus en profondeur d'autres domaines d'applications et d'autres corpus avec différentes caractéristiques. Par ailleurs, une de nos perspectives à court terme est d'évaluer la découverte d'attributs littéraux telle que définie dans l'approche et le module d'interrogation de REISA en testant différentes fonctions d'agrégation des poids associés aux faits.

Chapitre 8

Conclusion

Synthèse des contributions

Nous avons étudié dans cette thèse la problématique d'interrogation sémantique de bases de connaissances RDF publiées sur le Web sémantique et/ou issues de l'annotation de documents semi-structurés. Nous sommes partis du constat que les informations présentes dans ces bases de connaissances sont relativement peu nombreuses par comparaison au volume d'informations contenues dans les documents semi-structurés du Web pour proposer des approches permettant de trouver plus de réponses aux requêtes sémantiques grâce à l'annotation des documents. Nous nous sommes focalisés sur la tâche la plus difficile dans ce contexte, qui est celle de trouver des instances de relations sémantiques à partir des documents annotés.

Dans ce cadre, nous avons présenté un état de l'art sur les approches qui permettaient la découverte de telles relations à partir de documents semi-structurés. Dans un premier temps, nous avons proposé une modélisation homogène des différents éléments permettant d'intégrer les bases de connaissances et les annotations quelque soit leur provenance ou les outils utilisés pour les produire. Cette modélisation a été formalisée en RDF(S)/OWL à travers le modèle d'intégration *SIM* (cf. chapitre 3).

Dans un second temps nous avons présenté deux approches permettant de trouver de nouvelles réponses aux requêtes sémantiques formulées suivant une ontologie de domaine.

Notre première approche, SHIRI-Querying, permet de rechercher des parties de documents pertinentes référant aux instances de concepts et de relations recher-

chées par la requête sémantique de l'utilisateur.

Cette approche exploite des documents annotés par des concepts du domaine. Elle utilise les règles d'annotation proposées par [Thiam *et al.*, 2009] pour annoter les nœuds de document suivant l'hétérogénéité sémantique de leur contenu textuel. Les requêtes sémantiques des utilisateurs sont reformulées suivant une relation d'ordre que nous avons définie pour trier les nœuds de document à retourner à l'utilisateur. Cette relation d'ordre est fondée, d'une part, sur la notion de voisinage structurel des nœuds dans le document et, d'autre part, sur la composition sémantique des nœuds, explicité dans la phase d'annotation.

SHIRI-Querying a été expérimentée sur deux corpus réels extraits du Web. Les résultats de ces expérimentations montrent que l'approche permet effectivement de trouver des réponses constituées de nœuds de document contenant l'information recherchée. Ces tests ont également permis de constater que la relation d'ordre que nous avons définie a effectivement permis de trier les réponses finales suivant leur précision.

La deuxième approche que nous avons proposée, REISA, permet d'enrichir les bases de connaissances afin de retrouver de nouveaux faits au moment de l'interrogation. Cet enrichissement consiste à produire de nouvelles instances de relations sémantiques entre des instances de concept du domaine. Les instances de relations produites par enrichissement sont pondérées suivant la distance qui sépare les entités de documents qui ont permis leur production. Nous avons dans ce cadre proposé une approche d'interrogation qui permet d'intégrer les bases de connaissances issues de différentes sources (i.e. base de connaissances externes, provenant de l'annotation du corpus cible ou de l'enrichissement REISA). Cette interrogation tient compte des poids associés aux différentes bases de connaissances pour trier les réponses retournées par le moteur de recherche.

REISA a été expérimentée sur deux corpus extraits du Web. Un premier corpus d'appels à communication constitué de 511 documents HTML extraits de 32 sites d'appel à communication et un deuxième corpus constitué de 100 documents Wikipedia relatifs au domaine géographique. Les résultats montrent que *REISA* permet effectivement de trouver de nouvelles instances de relation et que les contrôles sémantiques effectués par les bases de connaissances et le Web (chapitre 6) permettent effectivement de filtrer les instances candidates en éliminer de nombreux cas d'erreur.

Discussion

Les approches que nous avons proposées dans cette thèse permettent de retrouver de nouvelles réponses à des requêtes sémantiques formulées suivant une ontologie de domaine dans deux cas de figure différents.

Dans le premier cas de figure, les annotateurs ne se sont pas intéressés à retrouver précisément les instances de concept et ont annoté les documents avec les concepts uniquement. Les requêtes sémantiques des utilisateurs n'auront ainsi aucune réponse car les instances de concepts ne sont pas identifiées et la recherche de relations entre instances ne peut pas aboutir. L'enrichissement par instances de relations n'est pas non plus envisageable à cause de l'absence d'instances de concepts.

Dans ce cadre, l'approche *SHIRI-Querying* constitue une bonne alternative qui rend possible de poser des requêtes sémantiques recherchant des relations sémantiques entre les instances en retournant des nœuds de documents pertinents au lieu des IRIs des instances demandés.

Cependant, l'approche *SHIRI-Querying* est plus adaptée aux domaines peu riches sémantiquement où le nombre de relations possibles pour le même couple de concepts est élevé. Cela est du au fait qu'en l'absence d'une base de connaissance, l'approche exploite uniquement les informations de domaine et de co-domaine des propriétés pour contrôler sémantiquement les nœuds de document pouvant référer à des instances de concept reliées par la propriété, bien que le tri par le voisinage structurel et la composition conceptuelle des nœuds permette d'améliorer la précision.

Dans le deuxième cas de figure, des annotations par instances de concepts ont pu être effectuées et incluent des instances qui ont une description dans des bases de connaissances préexistantes. Dans ce cas, nous avons étudié comment produire de nouvelles instances de relations sémantiques. Dans ce cadre, nous avons proposé l'approche *REISA* qui génère des instances de relations candidates en utilisant la proximité structurelle des entités de document annotés. Ces instances candidates sont ensuite contrôlées ou filtrées en exploitant tout d'abord les bases de connaissances disponibles. Ce contrôle est effectué par les bases de connaissances exploite les faits présents dans la base de connaissance, les axiomes de fonctionnalité (inverse) des propriétés et les règles spécifiques au domaine d'application si elles sont disponibles.

Un deuxième contrôle est effectué par le Web en soumettant des expressions textuelles des instances de relation candidates à un moteur de recherche Web afin de les valider. Ce contrôle via le Web est ici envisageable dans la pratique

car le contrôle préalable par les bases de connaissances a permis de diminuer largement le nombre d'instances de relations candidates. Par exemple, dans nos expérimentations nous sommes passés de 278 instances candidates générées par le voisinage sémantique des entités de document pour la relation *capitale-de* à seulement deux candidats en appliquant les règles du domaine et les autres filtres liés à la base de connaissances. Enfin, *REISA* pondère les instances de relations retenues par l'enrichissement avec une mesure de confiance différenciée qui permet (i) de les distinguer des faits des bases de connaissances préexistantes et (ii) de les trier suivant la proximité structurelle des entités de document qui ont permis de les générer.

L'approche *REISA* peut ainsi être appliquée sur tout corpus de documents semi-structurés dès que des instances de concepts décrites dans une base de connaissances ont pu être localisées. L'approche atteint cependant certaines limites si le contrôle se fonde principalement sur les appels au Web. Ce cas peut se produire si la description des instances de concept est pauvre dans la base de connaissances. C'est notamment le cas des instances créées à la volée à partir des entités de document dans le cas où les annotateurs n'ont pas pu trouver d'instances correspondant à ces entités dans les bases de connaissances préexistantes. Cette limite peut être aussi atteinte si le contrôle par la fonctionnalité (inverse) des propriétés et le filtrage par les règles du domaine ne permettent pas d'éliminer suffisamment de candidats erronés.

Cependant, REISA est complémentaire aux méthodes d'extraction « classiques » des relations sémantiques (e.g. approches à base de patrons lexico-syntaxiques, approches statistiques à base d'apprentissage). Elle peut être combinée à ces approches pour améliorer leur performance en termes de rappel et de précision grâce aux contrôles par les bases de connaissances et le Web (si l'approche classique utilisée ne les exploite pas).

Perspectives

À court terme nous envisageons d'effectuer des expérimentations plus approfondies des deux approches sur plus de documents Web et de relations cibles avec des ontologies et des bases de connaissances d'autres domaines.

Dans un second temps, si les premiers résultats de l'approche *REISA* se voient confirmés pour différents domaines et corpus, nous envisageons d'appliquer cette approche d'enrichissement sur tout le corpus Wikipedia et tester le passage à

l'échelle de l'approche.

Une troisième perspective à court terme est de tester différentes fonctions d'agrégation des fait pondérés dans l'approche d'interrogation de *REISA* et d'évaluer l'apport de cette pondération sur un ensemble de requêtes utilisateurs constituées de plusieurs patrons de triplets RDF.

Le fait de pouvoir dédier des graphes nommés différents aux relations extraites permet aussi de dissocier les connaissances suivant le degré de confiance ou suivant le processus employé pour leur extraction. Ainsi, il serait possible d'utiliser REISA pour assister des experts de domaine souhaitant peupler une ontologie en proposant automatiquement des faits candidats triés suivant leur poids. Dans ce contexte, une interface de validation peut exploiter les liens *sim:refersTo* pour visualiser les parties de documents référant aux instances de concepts ou aux littéraux.

Nous envisageons aussi d'interroger les bases de connaissances exploitées pour l'enrichissement à travers des endpoint SPARQL sans rassembler les bases de connaissances dans un entrepôt local. Dans ce cadre, les bases de connaissances ne sont pas forcément saturées suivant les règles d'inférence RDF(S)/OWL dont celles exploitant les alignements avec les autres ontologies.

Dans un tel cas, nous pouvons exploiter les alignements entre les ontologies et leurs instances au niveau de chaque base (e.g. requête présentée à la figure 8.1) ou effectuer les alignements en local avec un outil spécialisé (e.g. TaxoMap [Hamdi *et al.*, 2009] pour les ontologies, *LN2R* pour les instances de concept [Saïs *et al.*, 2009]). Des plans de requêtes adéquats devront être définis pour prendre en compte ces deux cas.

Il sera aussi intéressant d'étudier le cas mixte où certaines annotations du corpus ciblé utilisent uniquement des concepts alors que d'autres utilisent des instances de concepts décrites par des relations et des attributs dans une base de connaissances préexistantes (cf. figure 8.1. Dans un tel cas l'approche *REISA* peut être appliquée pour trouver de nouvelles instances de relations. Un algorithme de reformulation de requêtes devra être définit afin d'exploiter les bases de connaissances et les différents types d'annotation. Par exemple, la requête sémantique qui recherche des instances de concepts et de relations peut être progressivement reformulée en remplaçant les relations sémantiques une à une par des relations de voisinage entre nœuds de document indexés par des concepts. Une telle approche augmenterait la performance des approches *SHIRI-Querying* et *REISA* séparées et permettrait de pallier certaines de leurs limites actuelles. Pour cela, une nouvelle relation d'ordre devra être définie pour tenir compte des poids des triplets (générés par REISA) et de la composition conceptuelle des nœuds (explicitée par

SHIRI-Annot).

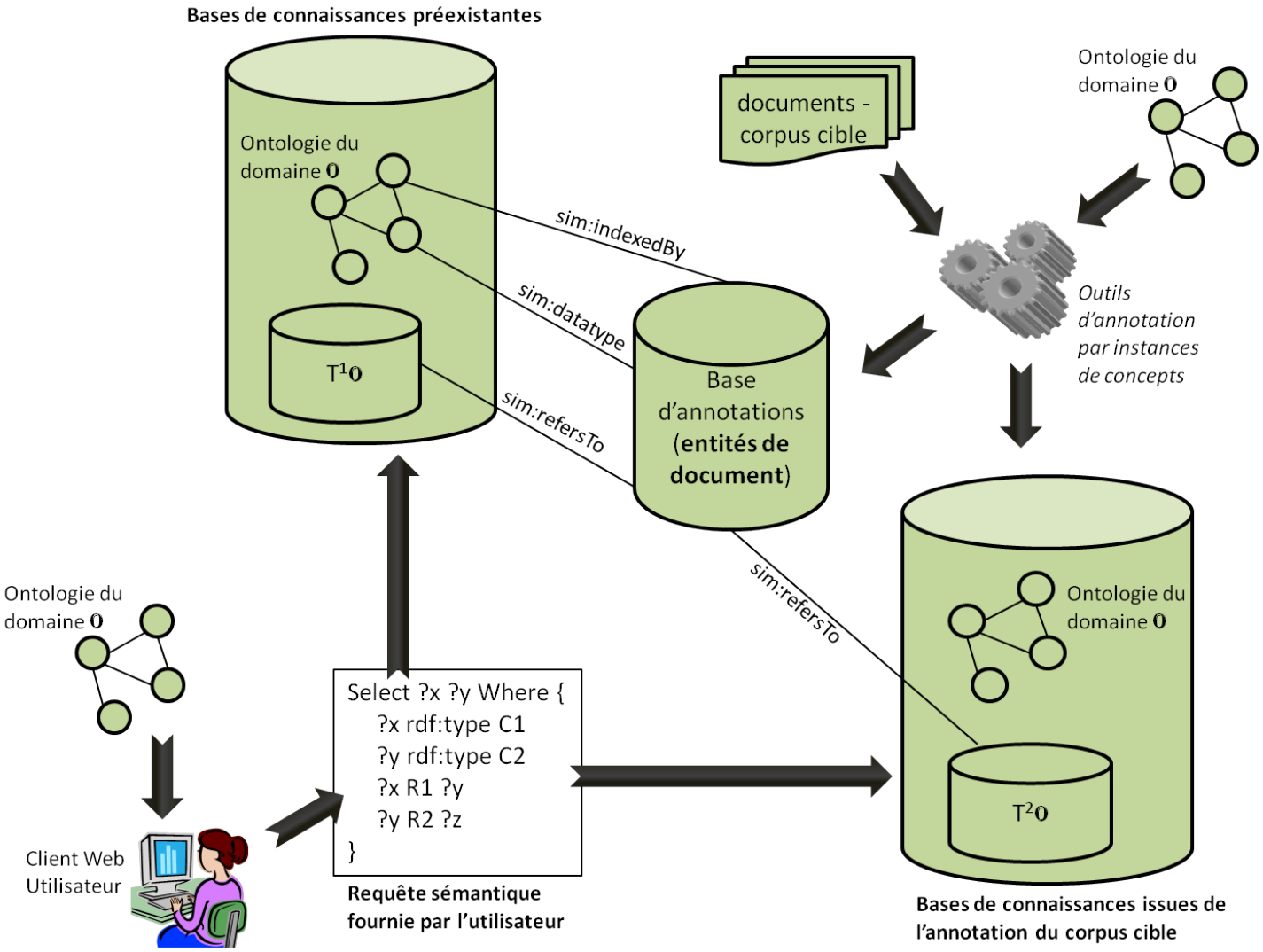

FIGURE 8.1 – Cadre mixte avec annotations par concepts et annotations par instance

Plusieurs autres perspectives sont envisageables comme applications de notre approche d'enrichissement. Par exemple, les instances de relations candidates éliminées par le critère de fonctionnalité peuvent être exploitées pour proposer des réconciliations entre instances de concept. À un autre niveau, un éditeur d'ontologie peut exploiter l'enrichissement que nous effectuons pour découvrir les parties de documents où des instances de concept proches ont été détectées sans qu'une instance de relation ne soit proposée par enrichissement. Cela permettra de trou-

ver de nouvelles relations à ajouter entre les concepts de l'ontologie.

Annexe A

Élements complémentaires

SHIRI Querying

Preuves

Dans le contexte des reformulations élémentaires appliquée à la requête Q_1 (cf. chapitre 4), nous avons défini trois types de patrons de sous-graphes de la requête :
– les singletons
– les patrons de sous-graphe connexes comparables
– les patrons de sous-graphe connexes non comparables
Dans ce qui suit, nous rappelons d'abord la composition de la requête Q_1 et les définitions nécessaires. Nous présentons ensuite les preuves des deux propriétés associées aux différents patrons de sous-graphe connexes de la requête.

La requête Q_1. La requête Q_1 est une transformation de la requête sémantique de l'utilisateur Q_0 recherchant des instances de concepts et de relations du domaine. Elle consiste à remplacer chaque variable référant à une instance de concept par une varibale $Node$, chaque relation de domaine par une relation *shiri:neighbor* et chaque attribut par le texte d'un nœud de document.

Concepts comparables. Deux concepts distincts sont dits comparables s'ils sont liés par la propriété *rdfs:subClassOf*.

Singleton. Un singleton, noté g_s, est un patron de sous-graphe connexe de P_1, contenant une seule variable de type $Node$.

Patron de sous-graphe connexe comparable. Un patron de sous-graphe connexe comparable de P_1, noté g_{cp}, est un patron de sous-graphe connexe de P_1 tel que les concepts de g_{cp}, représentés par l'ensemble $C(g_{cp})$, sont comparables.

Patron de sous-graphe connexe non comparable. Un patron de sous-graphe connexe non comparable de P_1, noté g_{cn}, est un patron de sous-graphe connexe de P_1 qui n'est pas comparable.

Propriété 1. Pour un partitionnement P_{ij} donné de la requête Q_1, l'intersection entre l'ensemble des patrons de sous-graphe connexes comparables et l'ensemble des patrons de sous-graphe connexes non comparables est égale à l'ensemble des singletons.

$$g_s = g_{cp} \cap g_{cn} \tag{A.1}$$

Dans le cadre de nos travaux, nous avons considéré uniquement les requêtes utilisateurs où les variables référant à des instances de concepts du domaine sont au plus typées par un concept du domaine.

La définition des concepts comparables est aussi uniquement valable pour les concepts distincts. Le fait de considérer les singletons comme étant des sous-graphes connexes comparables et non comparables est un choix de définition, effectué afin de permettre la reformulation des singletons en $PartOfSpeech$ et en $SetOfConcept$.

Propriété 2.

Tous patron de sous-graphe connexe est soit comparable soit non comparable.

$$g_c = \{g_{cp} \cup g_{cn}\} \tag{A.2}$$

Tout couple de concepts distincts d'un patron de sous-graphes connexe g_1, (C_1, C_2) est soit lié par la relation *rdfs:subClassOf* soit non lié.

Or par définition, s'il existe un couple de concepts distincts non lié par la relation *rdfs:subClassOf*, alors g_1 et non comparable et si tous les couples de concepts distincts de g_1 sont liés par la relation *rdfs:subClassOf*, alors g_1 est comparable.

Calculs de complexité

Dans ce qui suit, nous rappelons d'abord l'algorithme de partitionnement (cf. figure 2) et ces différentes notations. Nous présentons ensuite le calcul de la compexité au pire cas.

Notations :
- f_{pos} : Reformulation en $PartOfSpeech$.
- f_{set} : Reformulation en $SetOfConcept$.
- f_c : Reformulation en $Concept$.
- P_{ij} : Un partitionnement j ayant aboutit à i patrons de sous-graphes connexes.
- l_{ij} : nombre de singletons résultant du partitionnement P_{ij}.

Algorithme 2 Algorithme de reformulation DREQ

Début
  Pour $i \in 1 \dots n$ **Faire**
    Pour $P_{ij} : j \in 1 \dots j_{max}$ **Faire**
      $l^s_{ij} = \{g_{cp} de P_{ij}\}$
      $l^p_{ij} = \{g_{cn} de P_{ij}\}$
      `générer et exécuter les requêtes reformulées` $f_{set}^{l^s_{ij}} \circ f_{pos}^{l^p_{ij}} \circ f_c^{l_{ij}}$
      Pour $l \in (l_{ij} - 1) \dots 0$ **Faire**
        `générer et exécuter les requêtes reformulées` $f_{set}^{l^s_{ij}+l_{ij}-l} \circ f_{pos}^{l^p_{ij}} \circ f_c^{l}$
        `générer et exécuter les requêtes reformulées` $f_{set}^{l^s_{ij}} \circ f_{pos}^{l^p_{ij}+l_{ij}-l} \circ f_c^{l}$
Fin

Complexité au pire cas

Calculer la complexité au pire cas revient à calculer le nombre de reformulations pour une partition P_{ij} donnée et le nombre de partitionnements au pire cas.

Pour une partition P_{ij} donnée, les sous-graphes connexes comparables et non comparables ont une seule reformulation possible, f_{set} pour les premiers, f_{pos} pour les deuxièmes. Les singletons ont par contre trois reformulations possibles : f_c, f_{set} et f_{pos}. Le nombre de reformulations pour une partition P_{ij} correspond donc à :

$$\begin{aligned} \mathrm{N} &= \sum_{k=0}^{l_{ij}} C_{l_{ij}}^{k} \sum_{m=0}^{l_{ij}-k} C_{l_{ij}-k}^{m} \\ &= \sum_{k=0}^{l_{ij}} C_{l_{ij}}^{k} 2^{l_{ij}-k} \\ &= 3^{l_{ij}} \end{aligned}$$

Le nombre de toutes les partitions possibles pour la requête Q_1 est maximal si le patron de graphe RDF élémentaire de la requête utilisateur Q_0 est complet. Dans ce cas, le nombre de partitions correspond au nombre de Bell $B_n = \sum_{k=0}^{n-1} C_{n-1}^{k} B_k$ avec $B_0 = B_1 = 1$.

Liste des requêtes pour les expérimentations de *SHIRI-Querying*

Numéro	Requête
1	Select ?event ?eDate ?loc Where ?event rdf :type cfp :Event ?event cfp :hasLocation ?loc ?event cfp :hasDate ?eDate }
2	Select ?event ?top Where ?event rdf :type cfp :Event ?event cfp :hasTopic ?top ?event cfp :hasName ?eName FILTER(?eName ˜"IEEE") }
3	Select ?event ?loc ?top Where ?event rdf :type cfp :Event ?event cfp :hasTopic ?top ?event cfp :hasLocation ?loc ?event cfp :hasName ?eName FILTER(?eName ˜"IEEE") }
4	Select ?event ?loc ?top Where ?event rdf :type cfp :Event ?event cfp :hasTopic ?top ?event cfp :hasLocation ?loc ?event cfp :hasName ?eName FILTER(?eName ˜"Web Services") }
5	Select ?event ?mem ?top Where

?event rdf :type cfp :Event
?event cfp :hasTopic ?top
?event cfp :hasCommitteeMember ?mem
?event cfp :hasName ?eName }

Select ?event ?loc ?mem ?top Where
?event rdf :type cfp :Event
?event cfp :hasTopic ?top
?event cfp :hasCommitteeMember ?mem
6
?event cfp :hasName ?eName
FILTER(?eName ˜”machine”) }

Select ?event ?loc ?mem ?top Where
?event rdf :type cfp :Event
?event cfp :hasTopic ?top
?event cfp :hasLocation ?loc
7
?event cfp :hasCommitteeMember ?mem
?event cfp :hasName ?eName
FILTER(?eName ˜”machine”) }

Select ?event Where
?event rdf :type cfp :Event
8 ?event cfp :hasName ?eName
FILTER(?eName ˜”machine”) }

Select ?event ?mem Where
?event rdf :type cfp :Event
?event cfp :hasCommitteeMember ?mem
9
?event cfp :hasName ?eName
FILTER(?eName ˜”learning”) }

```
Select ?event ?eDate ?mem ?top Where
 ?event rdf :type cfp :Event
 ?event cfp :hasCommitteeMember ?mem
10 ?event cfp :hasDate ?eDate
 ?event cfp :hasName ?eName
 FILTER( ?eName ~"International") }

Select ?event ?eDate ?loc Where
 ?event rdf :type cfp :Event
 ?event cfp :hasLocation ?loc
11
 ?event cfp :hasDate ?eDate
 FILTER( ?eDate ~"2004") }

Select ?event ?date Where
 ?event rdf :type cfp :Event
12 ?event cfp :hasDate ?eDate
 FILTER( ?eDate ~"2006") }

Select ?event ?top Where
 ?event rdf :type cfp :Event
 ?event cfp :hasTopic ?top
13
 ?top cfp :hasName ?tName
 FILTER( ?tName ~"database") }

Select ?event ?top Where
 ?event rdf :type cfp :Event
 ?event cfp :hasTopic ?top
14
 ?top cfp :hasName ?tName
 FILTER( ?tName ~"web service") }
```

```
Select ?event ?top Where
  ?event rdf :type cfp :Event
  ?event cfp :hasTopic ?top
15
  ?top cfp :hasName ?tName
  FILTER( ?tName ~"machine learning") }
```

Bibliographie

Abdelhamid, E., Rafea, A. & El-Beltagy, S. (2009). Enhancing search results of concept annotated documents. In *Information Reuse Integration, 2009. IRI '09. IEEE International Conference on*, 330 –335. 12

Agichtein, E. & Gravano, L. (2000). Snowball : extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, DL '00, 85–94, ACM, New York, NY, USA. 18

Aussenac-Gilles, N. & Jacques, M.P. (2006). Designing and evaluating patterns for ontology enrichment from texts. In S. Staab & V. Svatek, eds., *Managing Knowledge in a World of Networks*, vol. 4248 of *Lecture Notes in Computer Science*, 158–165, Springer Berlin / Heidelberg, 10.1007/11891451_16. 17, 19

Berners-Lee, T., Hendler, J. & Lassila, O. (2001). The semantic web, publié en ligne : http ://www.sciam.com/article.cfm ?articleid=00048144- 10d2-1c70-84a9809ec588ef21. *Scientific American*. 1

Bhagdev, R., Chapman, S., Ciravegna, F., Lanfranchi, V. & Petrelli, D. (2008). Hybrid search : Effectively combining keywords and semantic searches. In *European Semantic Web Conference*. 26

Bikel, D.M., Miller, S., Schwartz, R. & Weischedel, R. (1997). Nymble : a high-performance learning name-finder. In *Proceedings of the fifth conference on Applied natural language processing*, ANLC '97, 194–201, Association for Computational Linguistics, Stroudsburg, PA, USA. 13

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R. & Hellmann, S. (2009). Dbpedia – a crystallization point for the web of data. *Web Semantics : Science, Services and Agents on the World Wide Web*, **7**, 154–165. 11, 20

Buche, P., Dibie-Barthélemy, J. & Hignette, G. (2008). Flexible querying of fuzzy rdf annotations using fuzzy conceptual graphs. In P. Eklund & O. Haemmerlé, eds., *Conceptual Structures : Knowledge Visualization and Reasoning*, vol. 5113 of *Lecture Notes in Computer Science*, 133–146, Springer Berlin / Heidelberg, 10.1007/978-3-540-70596-3_9. 21, 30

Buitelaar, P. & Siegel, M. (2006). Ontology-based information extraction with soba. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2321–2324. 21

Carroll, J.J., Bizer, C., Hayes, P. & Stickler, P. (2005). Named Graphs, Provenance and Trust. In *WWW '05 : Proceedings of the 14th international conference on World Wide Web*, 613–622, ACM, New York, NY, USA. 76

Castells, P., Fernàndez, M. & Vallet, D. (2007). An adaptation of the vector-space model for ontology-based information retreival. *IEEE Transactions on Knowledge and Data Engineering*, **19**, 20–27. 26, 27

Cimiano, P., Ladwig, G. & Staab, S. (2005). Gimme'the context : Context driven automatic semantic annotation with c-pankow. In *WWW conference*. 11, 13, 15, 29, 44

Corby, O., Dieng-Kuntz, R., Gandon, F. & Faron-Zucker, C. (2006). Searching the semantic web : Approximate query processing based on ontologies. *IEEE intelligent systems journal*, **21**, 20–27. 25

Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.M., Shaked, T., Weld, S.S.D. & Yates, A. (2004). Web-scale information extraction in knowitall. In *International World Wide Web conference (WWW)*. 18, 44

Gerber, D. & Ngomo, A.C.N. (2011). Bootstrapping the linked data web. In *1st Workshop on Web Scale Knowledge Extraction, International Semantic Web Conference (1)*, vol. 7031 of *Lecture Notes in Computer Science*, Springer. 18, 99

Grishman, R. & Sundheim, B. (1996). Message understanding conference-6 : a brief history. In *Proceedings of the 16th conference on Computational linguistics - Volume 1*, COLING '96, 466–471, Association for Computational Linguistics, Stroudsburg, PA, USA. 11

Gruber, T. (1993). A translation approach to portable ontologies. *Knowledge Acquisition*, **5**, 199–220. 10

Gruber, T. (1995). Toward principles for the design of ontologies used for knowledge sharing ? *International Journal of Human-Computer Studies*, **43**, 907–928. 10

Gruber, T. (2008). Ontology. In L. Liu & M.T. Özsu, eds., *the Encyclopedia of Database Systems*, Springer-Verlag. 10

Gulli, A. & Signorini, A. (2005). The indexable web is more than 11.5 billion pages. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, WWW '05, 902–903, ACM, New York, NY, USA. 1

Hamdi, F., Safar, B., Niraula, N. & Reynaud, C. (2009). TaxoMap in the OAEI 2009 alignment contest. In *The Fourth International Workshop on Ontology Matching*, Chantilly, Washington DC., États-Unis. 16, 109

Hassell, J., Aleman-Meza, B. & Arpinar, I. (2006). Ontology-driven automatic entity disambiguation in unstructured text. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold & L. Aroyo, eds., *The Semantic Web - ISWC 2006*, vol. 4273 of *Lecture Notes in Computer Science*, 44–57, Springer Berlin / Heidelberg, 10.1007/11926078_4. 13, 15

Hearst, M.A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics - Volume 2*, COLING '92, 539–545, Association for Computational Linguistics, Stroudsburg, PA, USA. 15

Hignette, G., Buche, P., Dibie-BarthÃ©lemy, J. & HaemmerlÃ©, O. (2009). Fuzzy annotation of web data tables driven by a domain ontology. In L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvanen, R. Mizoguchi, E. Oren, M. Sabou & E. Simperl, eds., *The Semantic Web : Research and Applications*, vol. 5554 of *Lecture Notes in Computer Science*, 638–653, Springer Berlin / Heidelberg, 10.1007/978-3-642-02121-3_47. 11, 21, 23

Hurtado, C.A., Poulovassilis, A. & Wood, P.T. (2006). A relaxed approach to rdf querying. In *International Semantic Web Conference, ISWC*. 25, 26, 27

Kamel, M. & Aussenac-Gilles, N. (2009). Utiliser la Structure du Document dans le Processus de Construction d'Ontologies (regular paper). In M.C. L'Homme & S. Szulman, eds., *Conférence Internationale sur la Terminologie et l'Intelligence Artificielle (TIA), Toulouse (France)*, (on line), IRIT, http ://www.irit.fr/. 17

Khelif, K. & Dieng-Kuntz, R. (2004). Ontology-based semantic annotations for biochip domain. In E. Motta, N. Shadbolt, A. Stutt & N. Gibbins, eds., *Engineering Knowledge in the Age of the Semantic Web*, vol. 3257 of *Lecture Notes in Computer Science*, 483–484, Springer Berlin / Heidelberg, 10.1007/978-3-540-30202-5_38. 11, 14

Limaye, G., Sarawagi, S. & Chakrabarti, S. (2010). Annotating and searching web tables using entities, types and relationships. *Proc. VLDB Endow.*, **3**, 1338–1347. 20

Mendes, P.N., Jakob, M., García-Silva, A. & Bizer, C. (2011). DBpedia spotlight : shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, 1–8, ACM, New York, NY, USA. 13, 15

Mrabet, Y., Pernelle, N., Bennacer, N. & Thiam, M. (2009). Aggregative and Neighboring Approximations to Query Semi-Structured Documents. In R. E-15, ed., *Actes de la conférence "Extraction et gestion des connaissances" - EGC'2009*, pp. 469–470, Cépaduès, Strasbourg, France, iSBN 978-2-85428-878-0. 3

Mrabet, Y., Bennacer, N., Pernelle, N. & Thiam, M. (2010a). Supporting semantic search on heterogeneous semi-structured documents. In B. Pernici, ed., *Advanced Information Systems Engineering*, vol. 6051 of *Lecture Notes in Computer Science*, 224–229, Springer Berlin / Heidelberg, 10.1007/978-3-642-13094-6$_1$8.3

Mrabet, Y., Bennacer, N., Pernelle, N. & Thiam, M. (2010b). Une approche pour la recherche sémantique de l'information dans les documents semi-structurés hétérogènes. In C. de Publication Universitaire 2010, ed., *COnférence en Recherche d'Infomations et Applications - CORIA 2010, 7th French Information Retrieval Conference, Sousse, Tunisia, March 18-20, 2010. Proceedings.*, 195–210, Sousse, Tunisie, fondation DIGITEO, projet SHIRI. 3

Mrabet, Y., Bennacer, N. & Pernelle, N. (2012). Enrichissement contrôlé de bases de connaissances à partir de documents semi-structurés annotés. In *Actes des 23es Journées Francophones d'Ingénierie des Connaissances- IC 2012*, no. ISBN 978-2-7466-4577-6 in IC 2012, 17–32, Paris, France. 4

Nadeau, D. & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, **30**, 3–26. 13, 16

Näppilä, T., Järvelin, K. & Niemi, T. (2008). A tool for data cube construction from structurally heterogeneous xml documents. *Journal of the American Society for Information Science and Technology*, **59**, 435–449. 22

Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D. & Goranov, M. (2004). Kim - semantic annotation platform. *Journal of Natural Language Engineering*, **10**, 375–392. 11, 13, 44

Saïs, F., Pernelle, N. & Rousset, M.C. (2009). Combining a logical and a numerical method for data reconciliation. In S. Spaccapietra, ed., *Journal on Data Semantics XII*, vol. 5480 of *Lecture Notes in Computer Science*, 66–94, Springer Berlin / Heidelberg, 10.1007/978-3-642-00685-2_3. 109

Sereno, B., Shum, S.B. & Motta, E. (2005). Claimspotter : an environment to support sensemaking with knowledge triples. In *Proceedings of the 10th international conference on Intelligent user interfaces*, IUI '05, 199–206, ACM, New York, NY, USA. 12

Shadbolt, N., Hall, W. & Berners-Lee, T. (2006). The semantic web revisited. *Intelligent Systems, IEEE*, **21**, 96 –101. 1

Suchanek, F.M., Ifrim, G. & Weikum, G. (2006). Combining linguistic and statistical analysis to extract relations from web documents. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*, 712, ACM Press, New York, USA. 13, 18

Suchanek, F.M., Kasneci, G. & Weikum, G. (2008). Yago : A large ontology from wikipedia and wordnet. *Web Semantics : Science, Services and Agents on the World Wide Web*, **6**, 203 – 217, <ce :title>World Wide Web Conference 2007Semantic Web Track</ce :title>. 11, 20, 77

Suchanek, F.M., Sozio, M. & Weikum, G. (2009). Sofie : a self-organizing framework for information extraction. In *Proceedings of the 18th international conference on World wide web*, WWW '09, 631–640, ACM, New York, NY, USA. 13, 14, 18, 23, 29

Thiam, M., Pernelle, N. & Bennacer, N. (2008). Contextual and Metadata-based Approach for the Semantic Annotation of Heterogeneous Documents. In *Proceedings of the 1st Workshop on Semantic Metadata Management and Applications (SeMMA 2008) at the 5 th European Semantic Web Conference (ESWC 2008)*, vol. 346, 16–28, Tenerife, Espagne, iSSN 1613-0073. 64

Thiam, M., Bennacer, N., Pernelle, N. & Lo, M. (2009). Incremental ontology-based extraction and alignment in semi-structured documents. In

S. Bhowmick, J. King & R. Wagner, eds., *Database and Expert Systems Applications*, vol. 5690 of *Lecture Notes in Computer Science*, 611–618, Springer Berlin / Heidelberg, 10.1007/978-3-642-03573-9_51. 4, 11, 12, 13, 29, 44, 45, 47, 49, 59, 64, 106

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 338–353. 85